인사말

신학기가 시작된 지 엊그제 같은데 벌써 옷깃을 여미게 되는 만추의 계절이 돌아온 것 같습니다. 유난히 장맛비가 많았던 금년도에는 각종 재해가 끊이질 않았던 것 같습니다. 이러한 어려운 여건 속에서도 수확의 계절이 어김없이 돌아왔습니다. 아울러 금년은 졸업을 앞둔 4학년 재학생들에게는 남다른 감회가 있는 해일 것입니다. 거의 1년 동안 졸업을 하기 위해 준비해 왔던 졸업 작품을 발표하고 전시하는 해이기 때문입니다. 완성도 높은 졸업 작품을 제작하기 위해 동분서주하고 많은 고생을 한 4학년 재학생 여러분의 노력에 치하를 보내 드립니다. 이러한 노력의 결과로 완성된 작품을 발표하고 전시하게 된 것은 오로지 여러분들의 능력을 마음껏 보여준 학창시절의 소중한 경험이기도 합니다. 이러한 소중한 경험은 여러분들이 졸업 후 사회에 진출하였을 때 엄청난 자산임을 명심하여야 합니다. 졸업 작품을 준비하면서 여러분들은 많은 것을 배웠을 것입니다. 주제선정, 자료수집방법, 자료 분석, 기존 보안제품에 대한 이해, 보안 프로그래밍 능력, 시스템 및 네트워크 구축에 대한 실무경험, 팀 프로젝트 수행에 대한 이해, 작품을 완성해 가는 업무처리 능력, 보고서 작성법, 그리고 발표자료 제작 등 다양한 경험들이 졸업 후 기업에 취업했을 때 많은 보탬이 되리라 확신합니다. 아무쪼록 이번 졸업 작품 전시회가 4년 동안 여러분들이 갈고 닦은 전공을 유감없이 발휘하는 장이 되었으면 하는 바램입니다. 또한 이번 전시회가 선 후배간의 지식과 경험을 교류하는 의미 있는 행사가 되길 기원하며 나아가 정보보호학과의 전통과 기틀을 다지는 장이 되었으면 합니다.

끝으로 이번 졸업 작품을 열정적으로 지도해 주신 각 지도교수님들께 그 노고를 치하하며 졸업 작품을 끝까지 준비하여 발표하게 된 4학년 여러분 모두의 노력에 박수를 보냅니다.

2010년 10월 19일

정보보호학과 학과장 양 정 모 교수

2010년 정보보호학과 졸업준비위원회

○ 위원장

최장섭 **홍희채**

○ 조별위원

1조 : **최장섭**, 홍희채, 김영주, 이재혁, 공대희, 조재승

2조 : **김태준**, 김재덕, 임우진, 김영식, 김선진

3조 : **최연규**, 최형진, 신기식, 이욱진, 박하나, 김선일

4조 : **서우영**, 구자연, 최혁수, 김면중, 최미진, 박일환

목 차

ㅇ 졸업작품

네트워크 보안 분야

• 실시간 네트워크 모니터링 시스템 관제(1조)

• Nessus와 Snort를 이용한 네트워크 구축(4조)

암호 기술 분야

• LFSR을 이용한 의사난수생성기 설계 및 구현(2조)

• 안드로이드 모바일 O.T.P 어플리케이션 및 시스템 구현(3조)

네트워크 보안 분야

• 실시간 네트워크 모니터링 시스템 관제(1조)

• Nessus와 Snort를 이용한 네트워크 구축(4조)

2010 졸업연구 결과보고서

실시간 네트워크 모니터링 시스템 관제

Real - Time Network Monitoring System

팀 명	CREATOR
지도교수	유 승 재 교수님
	박 인 숙 교수님
팀 원	최 장 섭(4학년)
	홍 희 채(4학년)
	조 재 승(4학년)
	김 영 주(4학년)
	이 재 혁(4학년)
	공 대 희(4학년)
스 텝	정 현 섭(3학년)

2010. 10

중부대학교 정보보호학과

요 약 문

1. 연구제목

실시간 네트워크 모니터링 시스템

Real - Time Network Monitoring System

2. 연구 목적 및 필요성

인터넷을 통해 확산되는 악의적인 소프트웨어 및 기업의 네트워크의 리소스에 대한 외부와 내부의 위협이 크게 증가 하고 있다.

기업 인프라에 대한 가장 큰 위협 중 기업의 내부 공격으로 인한 피해는 네트워크 관리자와 같이 신뢰도가 높은 위치에 있는 사람들의 행위에서 비롯된다.

전 세계 수많은 기관에서 규정을 강화함에 따라 기업에서는 네트워크를 모니터링하고 리소스 액세스 요청을 확인하고 네트워크에 로그온 및 로그오프 하는 사용자를 식별해야 할 필요성이 크게 확대되고 있다. 또한 기업들은 규정에 따라 특정 기간 동안 모니터링 한 보안 데이터를 보관해야 할 수도 있어야 된다.

3. 연구 내용

실시간 네트워크 모니터링 시스템은 네트워크를 실시간 적으로 모니터링 하여 올바른 네트워크 작동을 유지하는데 있어서 정보를 모니터링을 하는데 있어서 그 기반을 두는 시스템을 말한다. 또한 이 네트워크 모니터가 설치된 컴퓨터의 네트워크 어댑터로 전달되는 네트워크 Traffic 대한 정보 수집이 가능하며 해당 정보를 캡처 하고 분석 하여 여러 유형의 네트워크 문제를 방지하고 진단하며 해결할 수 있다.

모니터링 시스템을 통하여 각 네트워크 별 PC 리소스, 네트워크 패킷, 네트워크 트래픽 등 여러 가지 모니터링이 가능하며 이렇게 모니터링 된 자료들을 기반으로 외부로부터의 불법 침입자나 내부의 적을 감시 할 수 있다.

4. 연구 결과

실시간 네트워크 모니터링 시스템을 네트워크 환경에 구축 이를 적용시켜 침입방지 시스템 또는 침입차단 시스템의 방안까지 제안하는데 목적을 두고 연구를 진행한다

목 차

Ⅰ. 연구 계획

1 연구 목적

(1) 인터넷을 통해 확산되는 악의적인 소프트웨어 및 기업의 네트워크의 리소스에 대한 외부와 내부의 위협이 크게 증가 하고 있다.

(2) 기업 인프라에 대한 가장 큰 위협 중 기업의 내부 공격으로 인한 피해는 네트워크 관리자 와 같이 신뢰도가 높은 위치에 있는 사람들의 행위에서 비롯된다.

(3) 전 세계 수많은 기관에서 규정을 강화함에 따라 기업에서는 네트워크를 모니터링하고 리소스 액세스 요청을 확인하고 네트워크에 로그온 및 로그오프 하는 사용자를 식별해야 할 필요성이 크게 확대되고 있다. 또한 기업들은 규정에 따라 특정 기간 동안 모니터링 한 보안 데이터를 보관해야 할 수도 있다.

2 연구 개요

(1) 최근 네트워크 공격 후 마비나 네트워크 정보 탈취에 관하여 이에 대한 대응 분야가 정보보호기술의 중요한 부문으로 부각됨으로써 이에 따른 연구가 중요한 시점으로 떠오르고 있다.

(2) 따라서 실시간 네트워크 모니터링 툴을 이용하여 각종 Traffic 및 Packet을 분석 이를 연구 하여 정보보호기술의 응용력을 향상 및 대응력을 키우는 방법이다.

3 관련 연구

(1) **IDS (Intrusion Detection System) 침입탐지시스템**

침입 탐지 시스템은 전통적인 방화벽이 탐지할 수 없는 모든 종류의 악의적인 네트워크 Traffic 및 컴퓨터 사용을 탐지하기 위한 것으로 이것은 취약한 서비스에 대한 네트워크 공격과 Application 의 데이터 처리 공격 그리고 권한 상승 및 침입자 로그인 / 침입자에 의한 주요 파일 접근 / 컴퓨터 바이러스, 트로이 목마, Worm 과 같은 호스트 기반 공격을 포함한다.

(2) IPS (Intrusion Prevention System) 침입방지시스템

외부 네트워크로부터 내부 네트워크로 침입하는 네트워크 Packet을 찾아 제어하는 기능을 가진 소프트웨어 또는 하드웨어이다. 일반적으로 내부 네트워크로 들어오는 모든 Packet 이 지나가는 경로에 설치되며, 호스트의 IP주소, TCP/UDP의 포트번호, 사용자 인증에 기반을 두고 외부 침입을 차단하는 역할을 한다. 허용되지 않는 사용자 서비스에 대해 사용을 거부하여 내부 자원을 보호한다.

4 연구 방법

(1) 실시간 모니터링 시스템 구현 조사

최근 발생하는 Worm 바이러스나 서비스 거부 공격 등은 시스템과 네트워크의 구분이 모호할 정도로 상호 밀접하게 관련되어 있고, 시스템은 네트워크에 또한 네트워크는 시스템에 서로 영향을 주고 있으며 이러한 추세는 앞으로도 계속 진행될 것으로 보인다.

(2) 적용 가능한 시스템 기법 조사

- 각종 시스템 기법과 네트워크 모니터링 시스템 기법을 병합하는 방법을 조사한다.
- 연구 및 취약점 분석을 통해 기존 시스템에 강인성을 가지는 시스템 기법을 확정한다.
- 네트워크 모니터링을 프로그래밍에 반영하기 위한 프로그래밍 기법과 활용 가능한 함수 기능을 확인한다.
- 확정된 방법을 시스템으로 구현하기 위한 시스템 구성방안을 마련하고 실습환경을 구축 하고 파트 별 시스템 작업을 개시한다.
- 프로그래밍 작업과 병행하여 연구 결과를 IPS 및 IDS 에 접목시키는 방안을 연구, 최종 결과 보고에 반영한다.

5 최종 목표 및 결과의 활용

(1) 최종 목표

네트워크 모니터링을 통한 Traffic 및 Packet 분석으로 각종 해킹 및 공격 또는 네트워크의 비상 상황을 구현 경험 · 분석하여 앞으로 실제 상황에도 도움이 됨을 목적으로 한다.

(2) 결과의 활용

실시간 네트워크 모니터링 시스템 연구 결과를 네트워크 환경에 구축 이를 적용시켜 침입방지 시스템 또는 침입차단 시스템의 방안까지 제안하는데 목적을 두고 연구를 진행한다.

Ⅱ. 모니터링 시스템

1. RTNMS (Real Time Network Monitering System)

1-1 실시간 네트워크 모니터링 시스템

(1) 실시간 네트워크 모니터링 시스템은 네트워크를 실시간적으로 모니터링 하여 올바른 네트워크 작동을 유지하는데 있어서 정보를 모니터링을 하는데 있어서 그 기반을 두는 시스템을 말한다. 또한 이 네트워크 모니터가 설치된 컴퓨터의 네트워크 어댑터로 전달되는 네트워크 Traffic 대한 정보 수집이 가능하며 해당 정보를 캡처 하고 분석하여 여러 유형의 네트워크 문제를 방지하고 진단하며 해결할 수 있다.

(2) 모니터링 시스템을 통하여 각 네트워크별 PC 리소스 , 네트워크 패킷 , 네트워크 트래픽 등 여러 가지 모니터링이 가능하며 이렇게 모니터링 된 자료들을 기반으로 외부로부터의 불법 침입자나 내부의 적을 감시할 수 있다.

1-2 패킷 분석 및 트래픽의 종류

(1) 패킷 스니핑 또는 프로토콜 분석이라 불리며, 네트워크상에서 발생하는 일을 잘 이해하기 위해 네트워크상에 흐르는 실제 데이터를 수집하고 해석하는 과정을 말한다. 물리적인 라인을 통해 이동하는 네트워크 데이터를 수집하기 위해 사용되는 도구인 패킷 스니퍼를 통해 이뤄진다. 패킷 분석은 네트워크의 특징을 파악하고 네트워크를 누가 사용하고 있으며, 무엇이 대역폭을 많이 차지하는지, 네트워크 사용 피크 타임이 언제인지, 공격의 가능성이나 악의적인 활동들을 알아내고 비정상적인 어플리케이션을 찾아내는데 목적을 본다.

(2) 브로드 캐스트 트래픽

해당 포트가 허브, 스위치, 라우터인지에 상관 없이 네트워크 세그먼트에 있는 모든 포트로 데이터를 보낸다.

(3) 멀티캐스트 트래픽

한 컴퓨터에서 여러 대의 컴퓨터에 동시적으로 패킷을 전송하는 것을 말한다. 멀티캐스트의 목적은 가능한 작은 용량의 네트워크 대역폭을 이용해서 패킷을 전송하기 위한 것이다.

(4) 유니캐스트 트래픽

한 컴퓨터에서 다른 컴퓨터로 데이터를 직접 전송하는 방식이다.

(5) 브로드캐스트 도메인

다른 매체들을 통해서 연결된 다양한 허브나 스위치를 포함한 큰 네트워크에서는 전송된 브로드캐스트 패킷이 한 스위치에서 네트워크상에 있는 다른 스위치들에 있는 포트로 스위치에서 스위치로 재전송함으로써 무조건적으로 도달한다.

2. RTNMS 설치 및 구현

실제로 실시간 네트워크 모니터링 시스템을 운영하는데 있어서 시스템의 필요에 따라 다음과 같은 네트워크 환경을 구축한다.

① Web server - 공격을 받을 서버로써 일반적인 기업 형태의 웹 페이지를 구성 하고 있다.
② PRTG server - 각 서버들을 모니터링 하기 위한 서버
③ Log server - 로그 파일을 묶어서 관리자에게 전송하기 위한 서버
④ Wireshark server - 패킷을 탐지하고 캡처하기 위한 서버
⑤ Snort server - 네트워크 에서 패킷 탐지를 위한 서버

2-1 web server

(1) 웹서버는 클라이언트/서버 모델과 웹의 HTTP를 사용하여 웹 페이지가 들어 있는 파일을 사용자들에게 제공하는 프로그램이다.
가장 보편적인 웹 서버로는 32 비트 윈도우와 유닉스 기반의 운영체계에서 모두 쓸 수 있는 아파치와, 윈도우 NT에 딸려 나오는 IIS (Internet Information Server), 그리고 Netscape 의 Enterprise 서버 등이 있다.
1995년 NCSA 웹서버를 기반으로 탄생한 아파치는 ""open source"" 라이선스에 의거하여 배포되는 마음대로 쓸 수 있으며 현재 세계 웹서버 시장의 70%를 석권하고 있다.
IIS(Internet Information Server)는 마이크로소프트의 서버 운영체제인 Windows NT에 포함되어 나오고 있으며 NT에서만 사용 가능하다.

(2) webserver 의 필요한 패키지

① apache

아파치는 1995 년 그 당시에 가장 인기 있었던 웹 서버중의 하나인 NCSA HTTPD 1.3 버전을 기반으로 탄생하였습니다. 그 후 기존의 NCSA 웹 서버에 더욱 향상된 기능들을 탑재하여 Apache 웹 서버를 발표하였습니다. 현재는 인터넷 웹 서버 중에서 최고의 인기를 구가하고 있는 이른바 '잘 나가는' 소프트웨어 중의 하나다. 그 이유를 들자면 지속적으로 패치파일을 제공하고 최고의 퍼포먼스를 내고 있기 때문이다. 물론 무료로 제공된다는 점과 많은 Market Share 의 점유로 인하여 안정성을 인정받았다는 점도 한 이유가 된다. 그리고 Windows server 버전도 배포되고 있다.

② MySQL

MySQL은 600만 사용자를 확보하고 있는 다중 스레드, 다중 사용자 형식의 구조질의어 형식의 데이터베이스 관리 시스템(SQL DBMS)이다. MySQL AB가 관리 및 지원하고 있으며, Qt처럼 이중 라이선스가 적용된다. 하나의 옵션은 GPL이며, GPL 이외의 라이선스로 적용시키려는 경우 전통적인 지적재산권 라이선스의 적용을 받는다.

위와 같은 지원 방식은 자유 소프트웨어 재단이 프로젝트에 저작권을 적용하는 방법과 비슷한 JBoss의 모델과 유사하다. 그러나 기반코드가 개인의 소유자에게 저작권이 있고 커뮤니티에 의해 개발되는 아파치 프로젝트와는 다르다.

MySQL AB는 MySQL 라이선스에 의한 판매 지원 및 서비스 계약 시스템을 개발, 유지한다. 그리고 또한 인터넷을 통한 전 세계의 협력자들을 고용한다. MySQL AB는 David Axmark, Allan Larsson 그리고 Michael "Monty" Widenius에 의해 설립되었다.

MySQL AB는 또한 MaxDB라고 불리는 MySQL AB와는 기반코드가 다른 데이터베이스 관리 시스템을 판매하고 있다.

③ php

하이퍼텍스트 생성 언어(HTML)에 포함되어 동작하는 스크립팅 언어. 별도의 실행 파일을 만들 필요 없이 HTML 문서 안에 직접 포함시켜 사

용하며, C, 자바, 펄 언어 등에서 많은 문장 형식을 준용하고 있어 동적인 웹 문서를 빠르고 쉽게 작성할 수 있다. ASP(Active Server Pages)와 같이 스크립트에 따라 내용이 다양해서 동적 HTML 처리 속도가 빠르며, PHP 스크립트가 포함된 HTML 페이지에는 .php, .php3, .phtml이 붙는 파일 이름이 부여된다. 처음에는 'Personal Home Page Tools'이라 불렸으며, 공개된 무료 소스이다.

(3) web server 구축

① yum -y install httpd 명령어를 입력하여 apache를 RPM 설치 한다.
② yum -y install mysql-server 명령어를 입력하여 mysql을 RPM 설치 한다.
③ yum -y php 명령어를 입력하여 php를 RPM 설치 한다.
④ yum -y vsftpd 명령어를 입력하여 vsftpd를 RPM 설치 한다.
⑤ 주소창에서 http://localhost<그림 1-1>을 입력하여 apache 가 제대로 설치 되었는지 확인해본다.
⑥ vi 편집기를 이용하여 vi /var/www/html/phpinfo.php 파일을 편집한 후 http://localhost/phpinfo.php를 입력하여 php 가 제대로 설치 되었는지 확인한다.
⑦ 데이터베이스 관리를 위해 mysqld 데몬을 실행시켜서 초기 root 비밀번호를 설정 후 'zero' 라는 테이블을 생성한다
⑧ 웹 서버에서 기업의 홈페이지 구현을 위해 데이터를 업/다운로드 하기 위한 FTP 데몬으로 vsftpd를 실행한다.
⑨ FTP를 통하여 /var/www/html 위치에 홈페이지 파일들을 업로드 한다.
⑩ 파일들을 업로드 한 후의 웹서버의 웹 페이지 그림 참조
⑪ 부록 <소스 1>~<소스 1-9>, <그림 1>~<그림 1-5> 참조

2-2 PRTG server 구축

① PRTG 서버는 windows 2003 서버 버전에 맞춰서 설치를 한다.
② Setup 파일을 실행시켜서 설치를 시작한다.
③ PRTG 는 원래 상용화 제품이다.
④ 프리웨어 에디션 버전으로 설치 한다.
⑤설치 완료 후 기본 웹브라우저로 PRTG를 실행 시킨다.
⑥ 각 서버들마다 snmpd 데몬을 설치 한 후 /etc/snmpd/snmpd.conf 파일을 열어 안에 수정을 해준다.

```
========================= snmp.conf 설정 =========================
# First, map the community name "public" into a "security name"
#          sec.name            source            community
com2sec  local               localhost         public
com2sec  network             61.81.108.61      public

# Second, map the security name into a group name:
#          groupName         securityModel  securityName
group    rwgroup             v1         local
group    rwgroup             v2c        local
group    rwgroup             v1         network
group    rwgroup             v2c        network
####
# Third, create a view for us to let the group have rights to:
# Make at least  snmpwalk -v 1 localhost -c public system fast again.
#          name                incl/excl       subtree            mask(optional)
#view      systemview       included     .1.3.6.1.2.1.1
#view      systemview       included     .1.3.6.1.2.1.25.1.1
view      all                 included          .1         80
view      systemview          included          system
view      mib2                included          .iso.org.dod.internet.mgmt.mib-2 fc
####
# Finally, grant the group read-only access to the systemview view.
#          group              context sec.model sec.level prefix read    write   notif
#access    notConfigGroup ""          any            noauth       exact    systemview
     none  none
access   rwgroup ""         any       noauth   exact    all       all       all
access    rogroup ""          any        noauth    exact     systemview          none
     none
```

⑦ 이와 같은 방법으로 Ping - status 와 HTTP FULL PAGE 센서를 추가시켜준다.

⑧ 부록 그림 <그림 2>~<그림 2-11> 참조

2-3 Log Server

(1) 원격 로그 서버는 다른 시스템들의 log들을 저장할 하드 드라이브 공간을 제공하도록 미리 설정되어진 시스템일 뿐 그 이상도 그 이하도 아니다. 이 시스템은 완벽한 보안으로 차단되어 있어야 하며 모든 RPC 데몬 들이나 다른 기타 서비스들도 암호화되지 않고서는 절대 접근이 허락 되지 않는다. 데이터들은 오직 UDP/Port 514번을 통해서만 전송이 통제된다.

또한 이 로그서버를 통하여 쌓여 있는 로그 파일들을 토대로 서버의 접근 및 상태를 분석한다.

(2) 로그 분석

Linux 시스템은 다양한 로그를 남긴다.

특히 서버와 관련된 로그 파일은 관리에 있어서 대단히 중요한 역할을 한다.

시스템에 이상이 생겼을 때 혹은 보안이 뚫려서 해킹을 당했을 때 있어서 이에 대한 1차적인 확인을 로그 파일들에서 확인을 한다. 또한 어떤 문제점을 해결하는데 있어서 로그 파일은 중요한 역할을 담당한다.

Linux에서는 /var/log 디렉토리에 시스템의 모든 로그를 기록 및 관리하고 일반적으로 텍스트 형식으로 저장되어있다. 또한 시스템의 /etc/syslog.conf 파일에는 거의 모든 시스템 로그파일들의 위치를 저장하고 있다. vi, cat, less 간단한 문서편집를 통해 그 내용을 볼 수 있다. 그러나 일부 로그 파일의 경우엔 텍스트 형식이 dslsep , btmp 와 wtmp 가 바로 그것인데 이들은 각각 lastb 와 last 라는 명령어를 통해 그 내용을 확인할 수 있다. Linux 로그 파일은 일정 주기로 정리할 필요가 있다. 그렇지 않다면 어느 순간엔 로그 파일이 파일 시스템을 모두 차지하는 상황이 발생할 수 있다.

(3) 중요한 LOG 파일들

- boot.log : 리눅스가 부팅 될 때 뿌려주는 모든 메시지를 기록하고 있다. (부팅 에러 시 이 로그파일 참조)

- cron : 시스템의 정기적인 작업에 대한 로그를 남기고 기록하고 있다.

/etc/밑에 있는 파일들 중 cron.hourly, cron.daily, cron.weekly, cron.monthly 파일들은 각각 시간별 일별, 주별, 월별로 정기적으로 운영체제에서 자동으로 작업해야 할 것에 대한 작업을 저장 하고 있으며 지정한 일시에 실행이 되며 이들 작업을 한 후에는 /var/log/cron 파일에 기록을 남기게 된다.

- message : 운영체제에서 보내주는 실시간 로그를 관리하고 있으며, 주로 콘솔로 이 메시지는 실시간으로 보여준다.

- secure : 시스템의 접속에 관한 로그파일로서 언제, 누가, 어디에서 어떻게 접속을 하였는가에 대한로그를 기록하고 있다. 시스템에 불법 침입이 있었다고 의심이 될 때는 반드시 이 로그 파일을 확인해야 한다.

- xferlog : ftp로 로그인하는 사용자에 대한 로그를 기록하는 파일로서 /etc/ftpacess/에 그 설정 파일을 가지면 ftp의 홈디렉토리는 /home/ftp 이다.

(4) 로그 파일의 종류

- /dev/console - 콘솔에 뿌려지는 메시지들
- /var/log/messages - 모든 데몬의 시스템 로그
- /var/log/cron - crond 데몬 로그 파일
- /val/log/maillog - sendmail, pop등의 메일 관련 데몬의 로그
- /val/log/secure - 보안인증 관련 메시지
- /val/log/xferlog - ftp(proftpd,vsftpd) 로그
- /val/log/dmesg - 부팅시 각종 메시지들을 저장
- /val/log/wtmp - 시스템 전체 로그인 기록을 저장
- /val/log/utmp - 현재 로그인 사용자에 대한 기록, 사용자 ip 저장
- /val/log/lastlog - 현재 로그인 사용자에 대한 기록
- /val/log/spooler - (uucp, new 장치에서 위급상태(crit) 이상인 메시지 기록
- /val/log/httpd/access_log - 아파치 웹서버 로그들을 기록
- /val/log/httpd/error_log - 아파치 웹서버 에러들을 저장

(5) 시스템 각 계정의 최근 접속정보를 확인하는 log

- lastlog 는 /etc/passwd 파일에 정의 되어 있는 모든 계정의 최근

접속 정보를 확인하는 명령어 이다. 주로 서버의 보안점검을 위하여 필수적으로 확인해 보아야 하는 명령어로써 간단히 lastlog 라고만 하면 모든 계정의 마지막 접속정보를 출력해준다.

- 사용 방법은 lastlog [-u 계정명] [-t 일자]
- lastlog 는 /var/log/lastlog 파일의 정보에 저장된 정보를 참조하여 결과를 출력하며 /var/log/lastlog 파일은 바이너리파일로 되어 있기 때문에 cat이나 vI 등의 일반적인 방법으로는 확인할 수가 없다.
- /var/log/lastlog 파일에는 각 계정의 최근 접속 정보가 기록되는 파일로써 /usr/include/lastlog.h 파일에 정의된 포맷 형태로 로그파일에 저장된다.

(6) 특정 계정 사용자의 시스템 최근 접속정보 확인 하기

- 특정 계정만의 최근 접속방법을 확인하고자 한다면 -u 옵션과 함께 사용하면 된다.
  lastlog -u 계정명

(7) 지정한 최근까지의 시스템 접속정보 확인 하기

- lastlog 는 최근 일까지의 마지막접속정보를 확인 할 수도 있음, “last -t 일자 ” 와 같이 사용하면 지정된 일자까지의 접속한 최근 접속정보를 확인 할 수 있다.
  lastlog -t 일수

2-3-1 Log server 구축

① Log 서버는 web 서버로부터 로그를 받아 log 서버로 전송 후 log 서버에서 관리자 메일로 전송 하는 단계를 거치게 된다.
② 클라이언트 설정
③ 부록 그림 <그림 3>~<그림 3-8> 참조
④ 원격지 로그 서버
④ service syslogd stop 명령어로 데몬을 정지 시킨 후 다음과 같이 수정을 한다.

2-4. WireShark

(1) Wireshark

처음에 캔자스 주에 있는 미주리 대학교에서 제럴드 콤즈라는 사람에 의해 만들어졌다. 콤즈의 첫 번째 버전인 Ethereal 이라는 어플리케이션은 1998년 GNU라이센스에 의해 처음 배포됐다.

Ethereal 이 배포되고 8년 뒤 콤즈는 좀 더 나은 직장을 찾기 위해 그룹을 떠났다. 그리고 불행히도 그의 직원 중 한명이 Ethereal 이라는 이름의 모든 권리를 가지고 있었기 때문에 콤즈는 Ethereal 이라는 브랜드를 쓸 수 있는 모든 권한을 가지지 못했다. 그 대신 콤즈와 개발팀은 2006년 중반에 그 프로젝트를 Wireshark 라고 다시 이름 지었다.

(2) Wireshark 의 장점

지속적으로 많은 장점들을 제공하지만 매니아들과 패킷 분석전문가들에게 다양한 기능을 제공해 Wireshark를 계속 사용하고 개발하게 유혹 하고 있다.

(2)-1 지원되는 프로토콜

다양한 프로토콜을 지원하는 데 있어서 뛰어난데, 대략 850여개쯤 가능하다. 이 프로토콜들은 IP와 DHCP 같은 일반적인 것부터 AppleTalk 이나 BitTorrent 같은 좀 더 개인적인 프로토콜까지 모두 사용할 수 있다. Wireshark는 오픈 소스 형태이기 때문에 새로운 프로토콜이 생기면 쉽게 업데이트가 가능하다. Wireshark가 제공하지 못하는 프로토콜이 있을 경우 해당 내용을 직접 Wireshark 개발자들에게 보내면 해당 프로토콜을 추가할지 여부를 판단해서 피드백 해준다. 결국 Wireshark가 제공하지 못하는 프로토콜은 거의 없다고 생각하면 된다.

(2)-2 친 사용자 환경

인터페이스는 가장 쉬운 패킷 스니핑 어플리케이션 중 의 하나다. Wireshark는 GUI를 바탕으로 한 어플리케이션으로, 이해하기 쉬운 메뉴와 인터페이스를 제공한다. 사용성의 편의를 위한 다양한 특징도 가지고 있다. 예를 들어 프로토콜에 관련된 컬러 코딩과 실제 데이터의 자세한 그래픽 표시가 있다. 더 복잡하고 명령어 라인으로 동작하는 tcpdump 와 달리 Wireshark의 GUI을 제공한다.

(2)-3 비용

GPL의 원칙 아래 프리 소프트웨어로 제공되는 오픈 소스 프로그램이기 때문에 비용 면에서는 걱정할 필요가 없다.

(2)-4 프로그램 지원

프로그램 지원은 특별히 개별적으로는 지원해주지 않는다. Wireshark와 같은 무료로 배포되는 소프트웨어들은 Wireshark를 사용하는 사용자들이 상호 지원 해 주기 때문에 대부분 형식적인 지원은 없다. 와이어샤크의

사용자 그룹은 제일 활발하고 우수한 오픈 소스 프로젝트 중 하나이다.

(2)-5 운영체제 지원

Windows, Mac OS, Linux를 바탕으로 한 운영체제 등 거의 모든 운영체제를 지원해 준다.

2-4-1 wireshark 구축 <그림 4>~<그림 4-5> 참조

wireshark 구축은 매우 쉽다 홈페이지에 접속 후 다운로드 받아서 설정 없이 설치만 해주면 그만이다.

하지만 지금 같은 네트워크 환경에서는 Wireshark가 캡처하기엔 환경이 불안정하기에 데이터 패킷 캡처를 목적으로 arp 스푸핑을 사용하여 웹서버와 공격pc 사이에서 데이터를 캡처 해준다.

arp 스푸핑에 사용된 툴은 nemesis 란 툴을 사용하였다.

webserver 00:13:77:C7:E7:55 winxp 00:E0:7D:FA:15:F1 wireshark : 00-13-77-C7-E7-55

2-5. Snort server

(1) Snort 는 "sniffer and more"라는 말에서 유래되었는데, 처음 공개되었을 EO는 코드 도 얼마 되지 않는 단순한 패킷 스니퍼 프로그램이었다. 그러나 이후 현재의 IDS와 같이 rule을 이용한 분석 기능이 추가되고, 커뮤니티를 통하여 계속적인 기능 보완과 향상을 통해 지금과 같이 다양한 기능과 탁월한 성능을 갖춘 프로그램이 되었다.

(2) snort 는 오픈 소스를 개발 중인 패킷 캡처 라이브러리인 libpcap을 사용하여 패킷을 캡처하고, 수집된 패킷이 사전에 정의된 snort 공격 툴과 비교하여 만약 매칭 되었을 경우 syslog를 통해 로그를 남기거나 특정 디렉토리의 특정 파일 또는 database 에 남기도록 할 수 있다.

(3) Snort 의 구조

snort 프로그램은 몇 가지 구성 요소들이 플러그인 형태로 이루어져 있어 쉽게 각자의 환경에 따라 변경하고 수정할 수 있도록 되어 있는데, 기본적으로 다음과 같은 4가지 구성요소로 이루어져 있다.

(3)-1 스니퍼(Sniffer)
(3)-2 preprocessor (전처리기)
(3)-3 탐지엔진
(3)-4 로깅(출력)

(4) Snort 동작 구조

snort는 먼저 Sniffer를 통해 snort IDS를 통과하는 모든 패킷을 수집하게 된다. 여기에서 수집된 데이터는 바로 룰 기반의 탐지 엔진을 거치지 않고 그 전에 preprocessor를 통해 보다 효율적인 공격 탐지를 위해 HTTP encoding Plugin 이나 포트스캔 등 몇 가지 Plugin을 먼저 거치면서 매칭이 되는지 확인하게 된다. 물론 preporcessor 역시 모듈화 되어 있어 각자의 환경에 불편하다면 disable 할 수 있다. 예를 들면 RP 트래픽에 대해 탐지할 필요가 없다면 PRC 관련 preprocessor을 주석처리하면 된다.

그리고 preprocessor를 통과한 패킷은 snor IDS 의 핵심이라 할 수 있는 룰 기반의 탐지엔진을 거치면서 사전에 정의된 탐지 룰 과 매칭 되는지 확인하게 된다. 만약 룰에 매칭 되었을 경우에는 사전에 정의된 정책에 따라 로그에 남게 되고, 그렇지 않은 경우 통과를 하게 된다.

2-5-1 Snort server 구축

① 패키지 목록을 업데이트 해준다.

```
sudo apt-get update
```

② snort 와 mysql을 설치 해준다.

```
sudo apt-get install snort-mysql
sudo apt-get install snort
```

③ snort rule 자동 업데이트인 oinkmaster를 설치한다.

```
sudo apt-get install oinkmaster
```

④ oinkmaster 출력 디렉토리를 설정 해준다.

```
sudo oinkmaster -o /tmp/
ls /tmp
```

⑤ oinkmaster 를 snort rules 디렉토리에도 출력을 설정 해 준다.

```
sudo oinkmaster -o /etc/snort/rules/
```

⑥ snort.conf 파일을 설정 해 준다.

```
sudo vi /etc/snort/snort.conf
var HOME_NET any                    //snort 탐지 범위
var RULE_PATH /etc/snort/rules  //snort rule 변수 설정
```

⑦ snort 실행

```
sudo snort -c /etc/snort/snort.conf
```

⑧ snort 의 로그파일이 쌓이게 된다.

```
/var/log/snort/alert
```

⑨ mysql 과 연동을 위해 /etc/snort/snort.conf 파일을 설정한다.

512번째 라인에 주석을 달아준다.

output log_tcpdump: tcpdump.log

529번째 라인에 mysql db를 입력해 준다.

output database: log, mysql, user=root password=test dbname=db host=localhost

⑩ mysql 사용사 설정과 snor 유저 등록을 해준다.

```
mysql -u root
set password for root@localhost=password('snort');
create database snort;
grant insert,select on root.* to snort@localhost;
set password for snort@localhost=password('PASSWORD_SNORT_CONF');
grant create,delete,insert,select,update on snort.* to snort@localhost;
grant create,delete,insert,select,update on snort.* to snort;
exit
```

⑪ mysql 에 snort 테이블을 생성 해준다.

```
mysql -u root -p < ~/snort-2.3.2/schemas/create_mysql snort
```

⑫ mysql에 snort 테이블이 제대로 생성 되었는지 확인해 본다.

```
mysql -u root -p
show databases;
use snort
show tables;
+-------------------+
| Tables_in_snort   |
+-------------------+
| data              |
| detail            |
| encoding          |
| event             |
| icmphdr           |
| opt               |
| reference         |
| reference_system  |
| schema            |
| sensor            |
| sig_class         |
| sig_reference     |
| signature         |
```

```
| tcphdr              |
| udphdr              |
+-------------------+
16 rows in set (0.00 sec)
```

위와 같이 테이블이 생성되어있어야 한다.

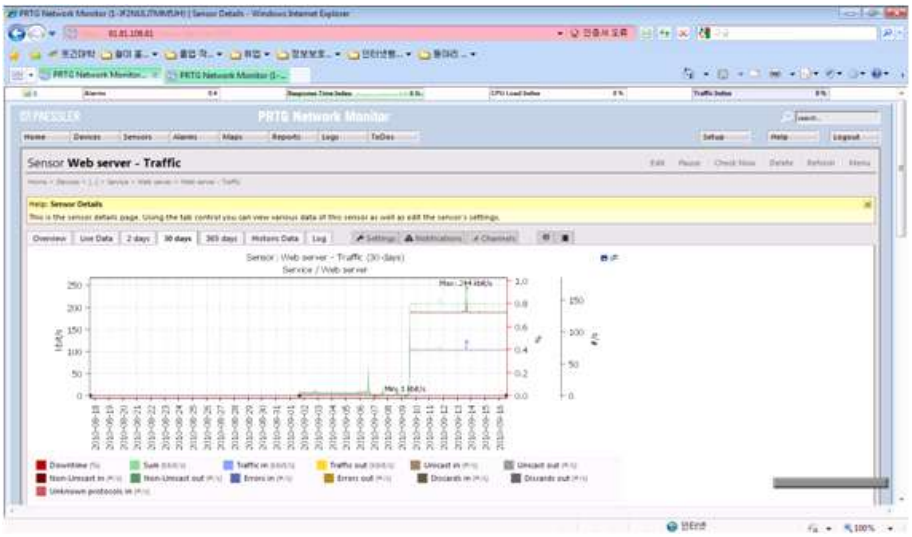
⑬ snort를 snort.conf 파일을 바탕으로 실행시킨다.

sudo snort -c /etc/snort/snort.conf

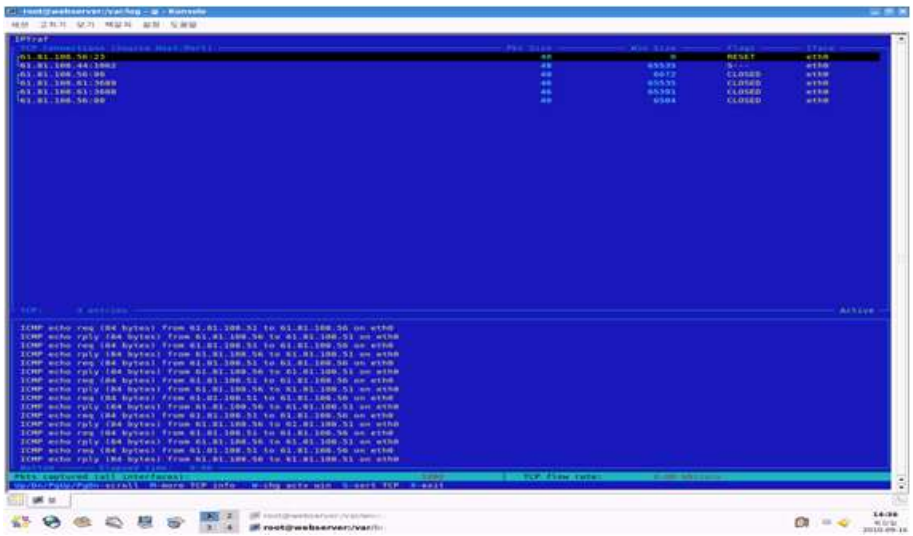
⑭ 구축된 Snort server 부록<그림 5>참조

Ⅲ. 연구 결과

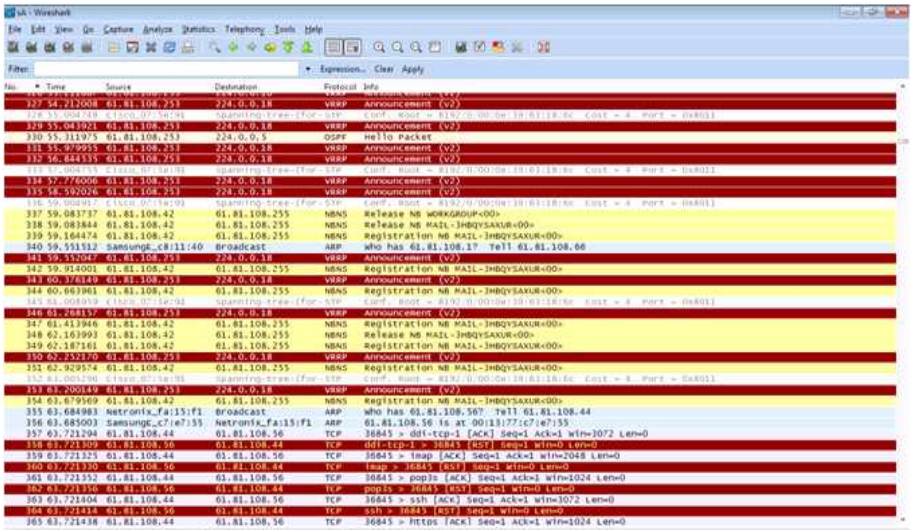
1. 실시간 네트워크 모니터링 상황 <그림 참조>



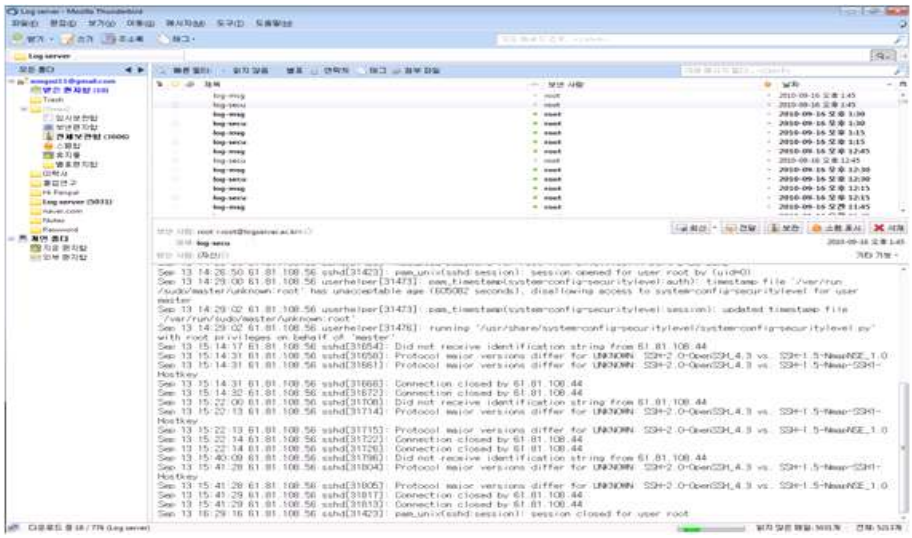
<그림 6> 실시간 Traffic 및 Status를 체크중인 PRTG 서버 http://61.81.108.61



<그림 7> IPtraf 로 실시간으로 간단히 캡처 되는 상황



<그림 8> 네트워크상의 패킷을 캡처하는 Wireshark



<그림 9> 매 15분마다 관리자로 전송되는 로그서버

2. 최종 연구 결과 보고

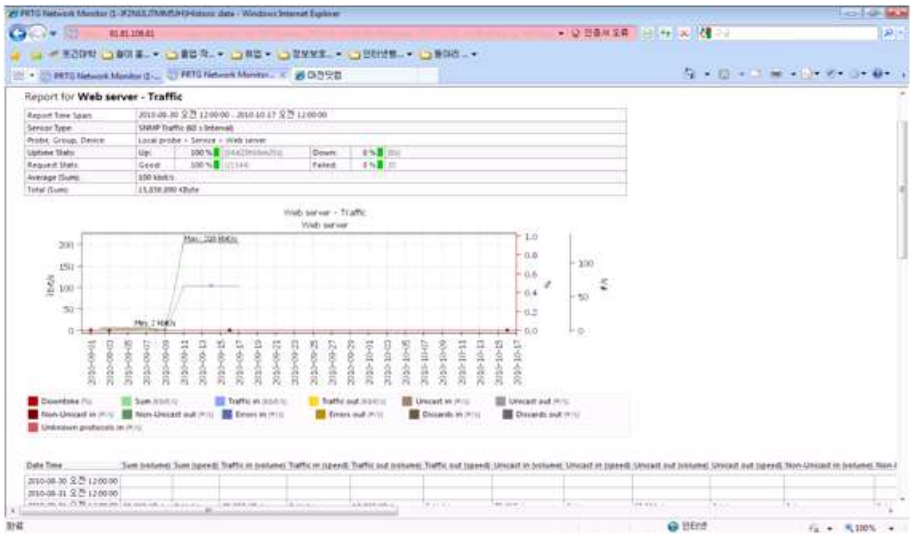
(1) 가상 시나리오에 따른 결과 보고

동기

한 사용자는 2010년 09월 11일 쇼핑몰 사이트 (마전닷컴) 에서 여러 가지 불만을 품고 악의 적인 목적을 가지고 접근을 시도하기 위하여 여러 가지 해킹 공격 방법 중에서 Telnet 과 SSH 접속을 시도 했다. 하지만 이 사용자는 Telnet, SSH 접근을 두 번 실패 한 후 포트 공격을 하기 위해 포트 스캔을 시작한다.

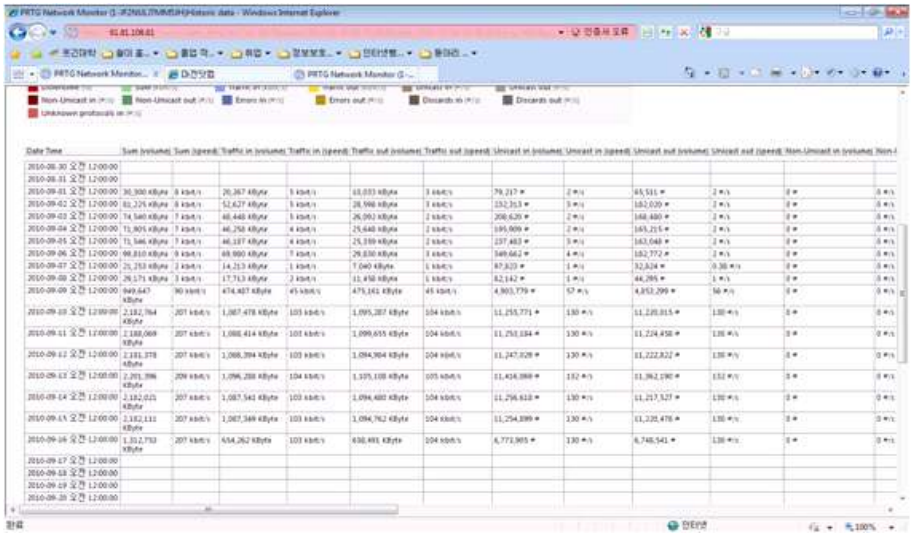
(2) 실시간 모니터링 시스템으로 탐지

먼저 관리자는 PRTG 모니터를 모니터링 하는 과정에서 일반적인 트래픽을 유지하던 그래프가 특정 수치로 상승하게 되고 이를 의심된다 여겨 데이터 분석을 시작한다.

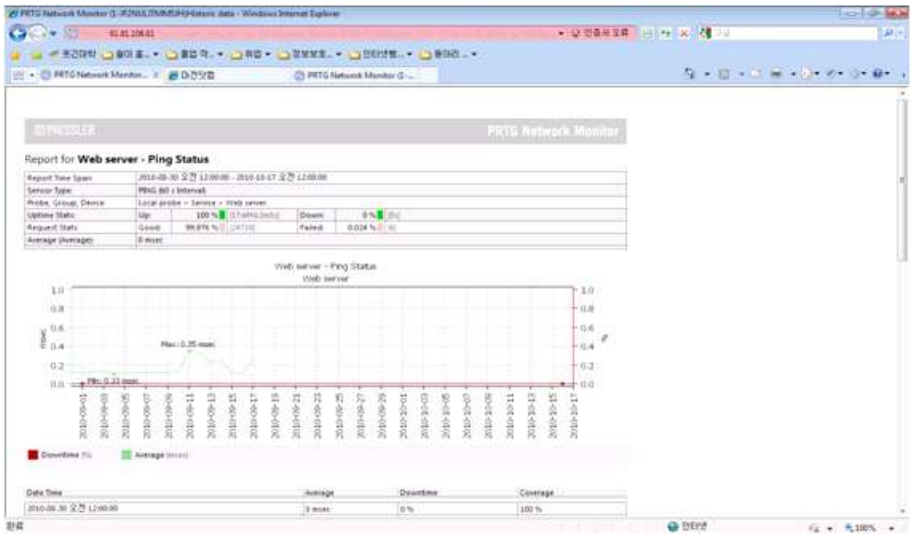


<그림 10> Traffic 그래프 확인 (2010 - 09 - 11)

<그림 10>의 그래프를 보면 일정 Traffic 을 유지하던 그래프가 09월11일을 기점으로 상승 곡선으로 올라간 후 일정량 그래프를 유지하고 있다. 이 부분은 사용자가 많다면 당연 트래픽이 많이 쌓이겠지만 앞부분의 낮은 사용량의 Traffic 을 보면 누군가 접근을 시도 하고 있음을 알 수 있다.



<그림 11> 역시 2010년 09월 10일 오전 10시부터 트래픽의 합계가 증가하였고 (29,171 KByte -> 949,647KBye) 속도나 Traffic In, Out 모두 변화가 생긴 것을 확인할 수 있다.

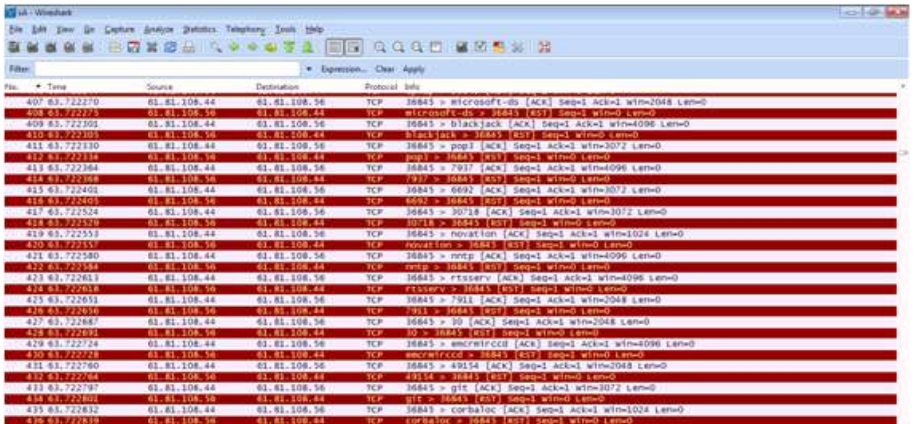


<그림 12> Ping Status 그래프 (2010-09-11)

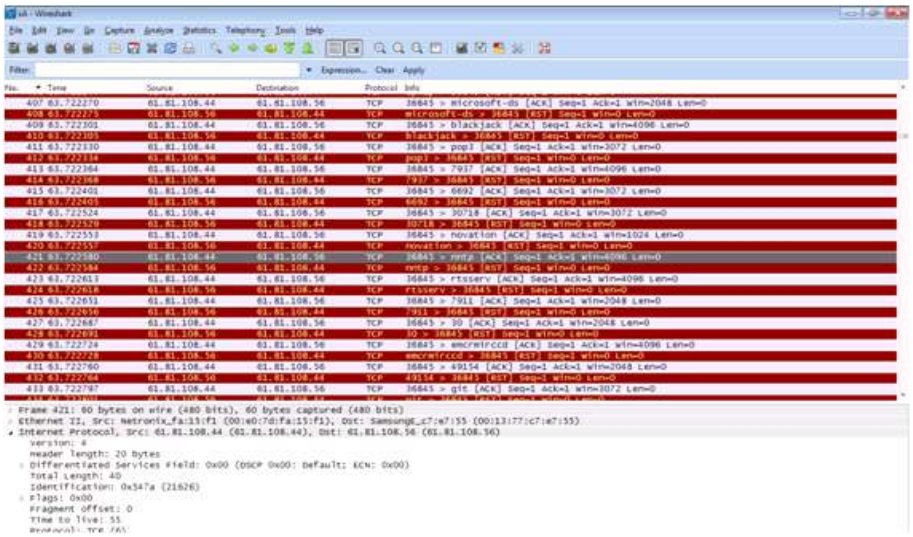
<그림 12>는 PING STATUS 그래프로써 역시 11일을 기점으로 그래프의 변화가 생겼다.

실시간적으로 캡처중인 Wireshark를 살펴보니 다음처럼 사용자가 (61.81.108.44)에서 웹서버 (61.81.108.56)으로 각 포트별로 패킷을 보내서 응답을

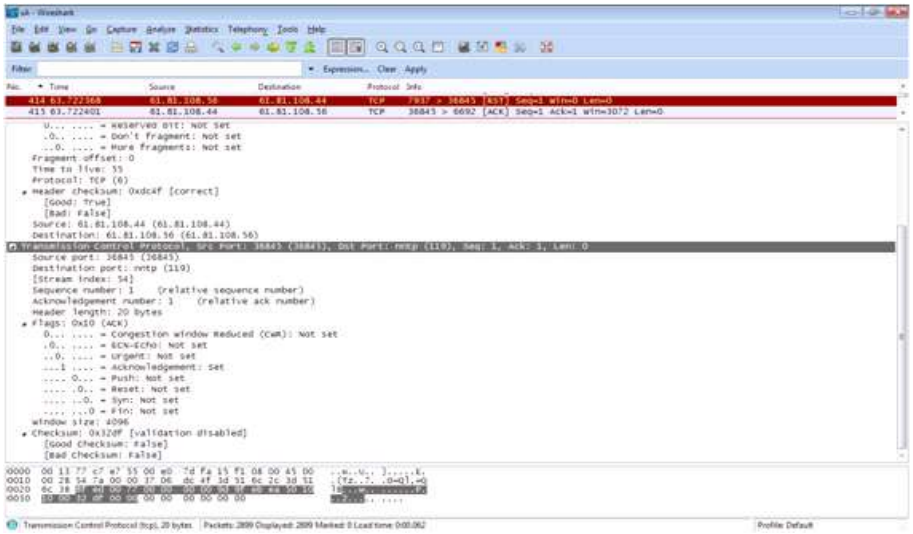
확인하는 방식의 포트 스캔을 했음을 알아 낼 수 있었다.



<그림 13> 61.81.108.41에서 61.81.108.56으로의 포트스캔 내용중 일부분을 분석해보았다.



<그림 14> TCP 프로토콜을 사용하였으므로 TCP부분을 분석한다.



<그림 15> 송신자는 36845번 포트를 사용해서 수신지 포트의 nntp(119)번 포트로 접근을 시도 했다.

요청번호는 1번으로 보냈고 응답번호 역시 1번으로 받았다. 헤더 길이는 20바이트가 사용되었다.

플래그 부분

0 . . . . . = Congestion windows Reduced (CWR) : Not set

. 0 . . . . . = ECN-Echo : Not set

. . 0 . ... = Urgent : Not Set

. . . 1 .... = Acknowledgement : Set

. . . . 0 . . . = Push : Not Set

. . . . 0 . . = Reset : Not Set

. . . . 0 . = Syn : Not Set

. . . . 0 = Fin : Not Set

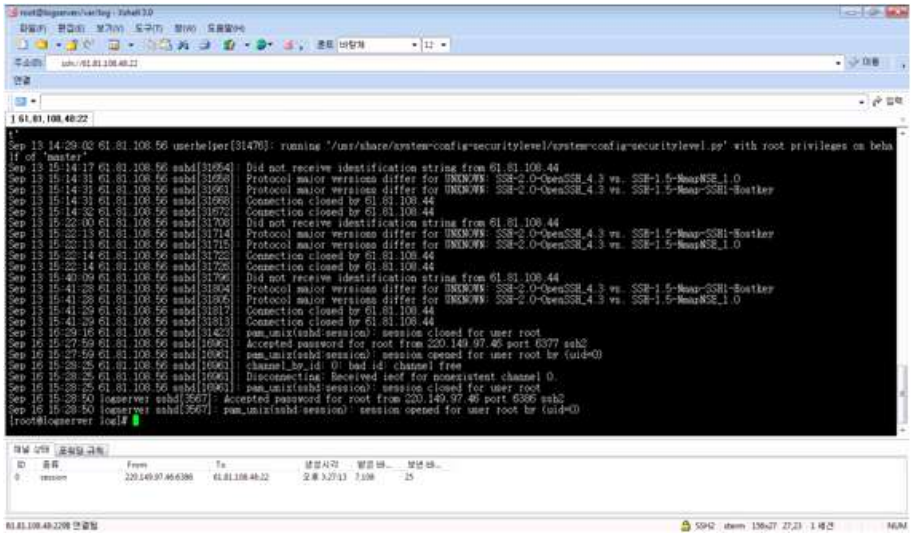
.....

플래그 부분을 보면 Acknowledgement 부분이 Set 되어 있는 것을 확인할 수 있다. 이것은 응답을 했다는 뜻이다.

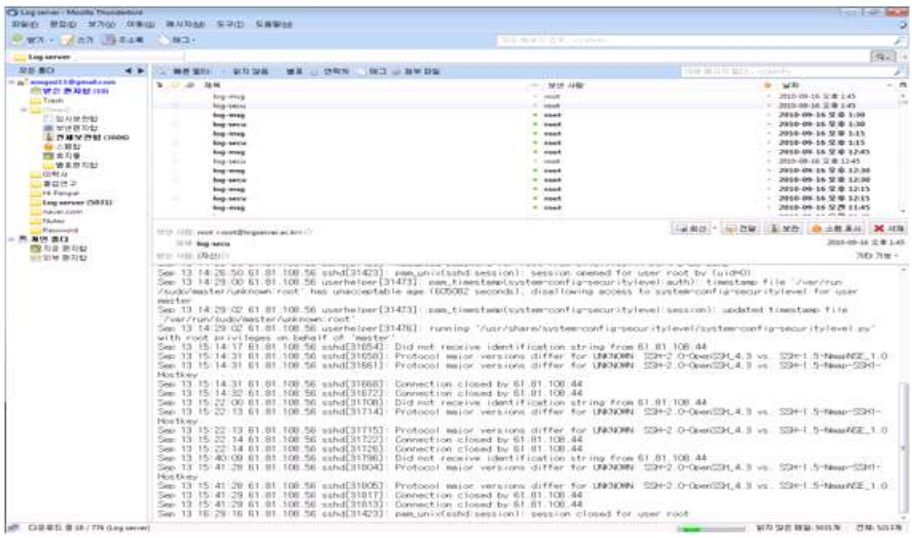
윈도우 사이즈는 4096 이다.

마지막으로 패킷의 오류를 점검하는 체크섬 (합계) 부분으로는 문제가 없음을 알 수 있었다.

관리자는 위의 모니터링을 통해서 로그 파일을 확인하게 된다.



<그림 16> /var/log/secure 원격 접속으로 웹 서버에 로그인 한 후 로그 파일을 열어 본 결과 61.81.108.56 서버로 61.81.108.44 의 ip가 접근했음 을 확인할 수 있다.



<그림 17> 로그 서버로부터 관리자 메일로 전송된 로그 파일이다. 원격 접속이 아니더라도 이 메일로 전송된 로그 파일로부터 분석이 가능하다.

3. 실시간 네트워크 모니터링 시스템을 만들어보고 난 이후 ..

졸업 작품으로 무엇을 할까 고민을 하던 우리 조는 최근 들어 DOS나 DDOS 공격이 활발히 이루어지는 것을 보고 이것을 미리 사전에 방지 하고 대응해 봄 으로써 목표를 설정했었지만 실력부족 상 직접 DOS 나 DDOS 는 써보지 못 했던 점이 매우 아쉬웠다. 또한 Snort 로는 탐지만이 가능하였지만 Snort를 조사 할 때는 많은 규칙이 미리 정의 되어 있고 이 정의되어 있는 규칙대로 여러 가지 해킹 기법을 탐지 해낼 수 있다는 것을 알 수 있었다. 그리고 여러 가지 툴들을 사용하여서도 강력하지는 않지만 어느 정도 시스템 구축이 가능하다는 것도 알 수 있었다. 이러한 툴들을 잘 조합하고 관리만 잘 해준다면 무자본으로도 시스템 구축이 가능하다.

비록 이번 졸업 작품은 실시간 네트워크 모니터링 시스템이었지만 이다음으로는 정말 중요한 IDS 와 IPS 도 역시 해볼 수 있도록 노력할 것이다.

Ⅳ. 부　록

```
root@localhost:~
파일(F)  편집(E)  보기(V)  터미널(T)  탭(B)  도움말(H)
 Package            Arch        Version                          Repository      Size
================================================================================
Updating:
 httpd              i386        2.2.3-43.el5.centos.3            updates         1.2 M
Updating for dependencies:
 httpd-manual       i386        2.2.3-43.el5.centos.3            updates         814 k
 mod_ssl            i386        1:2.2.3-43.el5.centos.3          updates          91 k
 openssl            i686        0.9.8e-12.el5_4.6                base            1.4 M

Transaction Summary
================================================================================
Install       0 Package(s)
Update        4 Package(s)
Remove        0 Package(s)

Total download size: 3.5 M
Downloading Packages:
(1/4): mod_ssl-2.2.3-43.el5.centos.3.i386.rpm                 |  91 kB     00:00
(2/4): httpd-manual-2.2.3-43.el5.centos.3.i386.rpm            | 814 kB     00:00
(3/4): httpd-2.2.3-43.el5.centos.3.i386.rpm                   | 1.2 MB     00:00
(4/4): openssl-0.9.8e-12.el5_4.6.i686.rpm                     | 8.0 kB     00:00
http://ftp.daum.net/centos/5.5/os/i386/CentOS/openssl-0.9.8e-12.el5_4.6.i686.rpm: [Errno 4] Sock
et Error: timed out
Trying other mirror.
(4/4): openssl-0.9.8e-12.el5_4.6.i686.rpm                     | 1.4 MB     00:00
--------------------------------------------------------------------------------
Total                                                 54 kB/s | 3.5 MB     01:06
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating       : openssl                                                      1/8
```

<소스 1>

```
root@localhost:~
파일(F)  편집(E)  보기(V)  터미널(T)  탭(B)  도움말(H)
Downloading Packages:
(1/4): mod_ssl-2.2.3-43.el5.centos.3.i386.rpm                 |  91 kB     00:00
(2/4): httpd-manual-2.2.3-43.el5.centos.3.i386.rpm            | 814 kB     00:00
(3/4): httpd-2.2.3-43.el5.centos.3.i386.rpm                   | 1.2 MB     00:00
(4/4): openssl-0.9.8e-12.el5_4.6.i686.rpm                     | 8.0 kB     00:00
http://ftp.daum.net/centos/5.5/os/i386/CentOS/openssl-0.9.8e-12.el5_4.6.i686.rpm: [Errno 4] Sock
et Error: timed out
Trying other mirror.
(4/4): openssl-0.9.8e-12.el5_4.6.i686.rpm                     | 1.4 MB     00:00
--------------------------------------------------------------------------------
Total                                                 54 kB/s | 3.5 MB     01:06
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating       : openssl                                                      1/8
  Updating       : httpd                                                        2/8
  Updating       : httpd-manual                                                 3/8
  Updating       : mod_ssl                                                      4/8
  Cleanup        : httpd-manual                                                 5/8
  Cleanup        : httpd                                                        6/8
  Cleanup        : openssl                                                      7/8
  Cleanup        : mod_ssl                                                      8/8

Updated:
  httpd.i386 0:2.2.3-43.el5.centos.3

Dependency Updated:
  httpd-manual.i386 0:2.2.3-43.el5.centos.3         mod_ssl.i386 1:2.2.3-43.el5.centos.3
  openssl.i686 0:0.9.8e-12.el5_4.6

Complete!
[root@localhost ~]#
```

<소스 1-1>

```
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
 * updates: ftp.daum.net
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package mysql-server.i386 0:5.0.77-4.el5_5.3 set to be updated
--> Processing Dependency: mysql = 5.0.77-4.el5_5.3 for package: mysql-server
--> Processing Dependency: perl-DBD-MySQL for package: mysql-server
--> Running transaction check
---> Package mysql.i386 0:5.0.77-4.el5_5.3 set to be updated
---> Package perl-DBD-MySQL.i386 0:3.0007-2.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

=================================================================================
 Package                    Arch            Version                     Repository          Size
=================================================================================
Installing:
 mysql-server               i386            5.0.77-4.el5_5.3            updates            9.8 M
Installing for dependencies:
 perl-DBD-MySQL             i386            3.0007-2.el5                base               148 k
Updating for dependencies:
 mysql                      i386            5.0.77-4.el5_5.3            updates            4.8 M

Transaction Summary
=================================================================================
Install       2 Package(s)
Update        1 Package(s)
Remove        0 Package(s)

Total download size: 15 M
Downloading Packages:
(1/3): perl-DBD-MySQL-3.0007-2.el5.i386.rpm                          | 148 kB     00:00
(2/3): mysql-5.0.77-4.el5_5.3. (10%) 30% [=======                ] 1.2 MB/s | 1.4 MB     00:02 ETA
```

<소스 1-2>

```
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
Dependencies Resolved

=================================================================================
 Package                    Arch            Version                     Repository          Size
=================================================================================
Installing:
 mysql-server               i386            5.0.77-4.el5_5.3            updates            9.8 M
Installing for dependencies:
 perl-DBD-MySQL             i386            3.0007-2.el5                base               148 k
Updating for dependencies:
 mysql                      i386            5.0.77-4.el5_5.3            updates            4.8 M

Transaction Summary
=================================================================================
Install       2 Package(s)
Update        1 Package(s)
Remove        0 Package(s)

Total download size: 15 M
Downloading Packages:
(1/3): perl-DBD-MySQL-3.0007-2.el5.i386.rpm                          | 148 kB     00:00
(2/3): mysql-5.0.77-4.el5_5.3.i386.rpm                               | 4.8 MB     00:02
(3/3): mysql-server-5.0.77-4.el5_5.3.i386.rpm                        | 9.8 MB     00:04
---------------------------------------------------------------------------------
Total                                                   1.9 MB/s |  15 MB     00:07
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating        : mysql                                                         1/4
  Installing      : perl-DBD-MySQL                                                2/4
  Installing      : mysql-server                                                  3/4
```

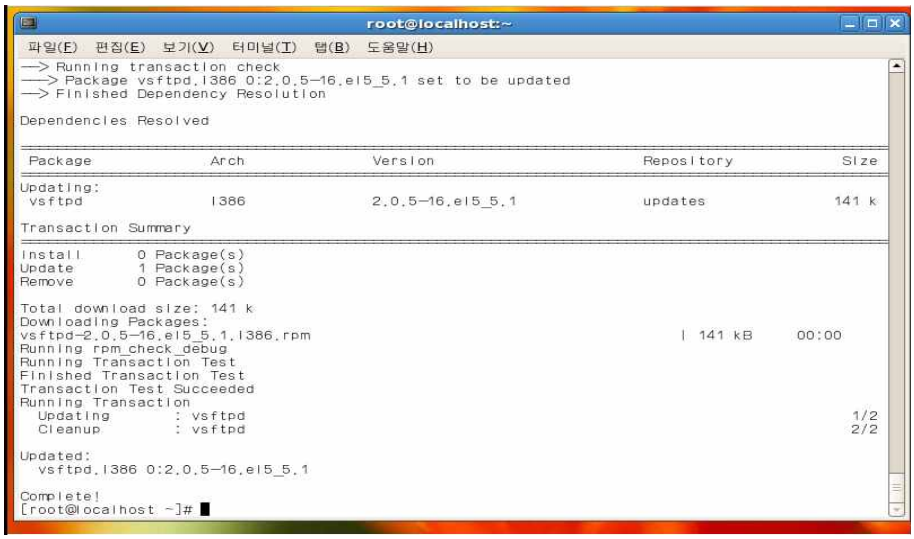
<소스 1-3>

```
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
---> Package php.i386 0:5.1.6-27.el5 set to be updated
--> Processing Dependency: php-cli = 5.1.6-27.el5 for package: php
--> Processing Dependency: php-common = 5.1.6-27.el5 for package: php
--> Running transaction check
---> Package php-cli.i386 0:5.1.6-27.el5 set to be updated
--> Processing Dependency: php-common = 5.1.6-23.2.el5_3 for package: php-ldap
---> Package php-common.i386 0:5.1.6-27.el5 set to be updated
--> Running transaction check
---> Package php-ldap.i386 0:5.1.6-27.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

=================================================================================
 Package              Arch          Version                  Repository      Size
=================================================================================
Updating:
 php                  i386          5.1.6-27.el5             base            2.3 M
Updating for dependencies:
 php-cli              i386          5.1.6-27.el5             base            2.1 M
 php-common           i386          5.1.6-27.el5             base            152 k
 php-ldap             i386          5.1.6-27.el5             base             37 k

Transaction Summary
=================================================================================
Install       0 Package(s)
Update        4 Package(s)
Remove        0 Package(s)

Total download size: 4.6 M
Downloading Packages:
(1/4): php-ldap-5.1.6-27.el5.i386.rpm                          |  37 kB     00:00
(2/4): php-common-5.1.6-27.el5.i386.rpm                        | 152 kB     00:00
(3/4): php-cli-5.1.6-27.el5.i3 (18%) 30% [=======          ] 642 kB/s | 664 kB     00:02 ETA
```

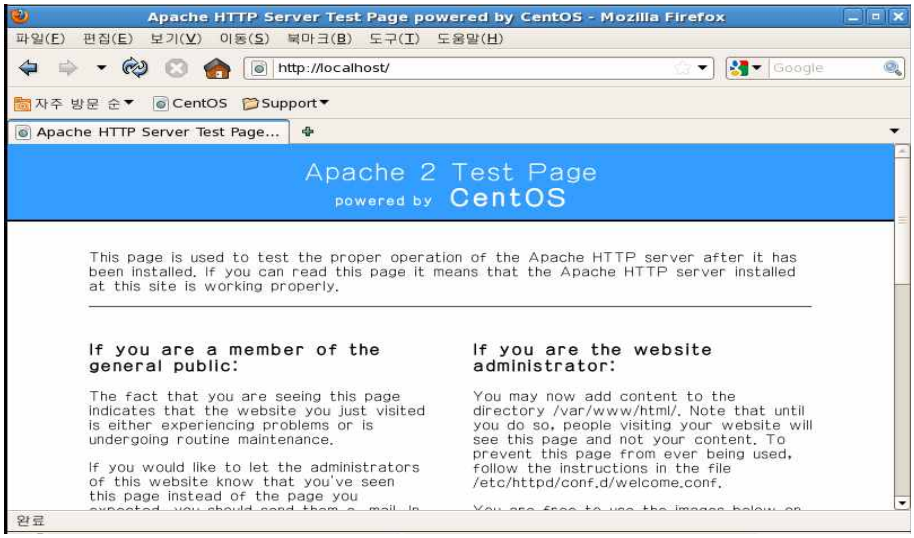
<소스 1-4>

```
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
Loading mirror speeds from cached hostfile
 * addons: ftp.daum.net
 * base: ftp.daum.net
 * extras: ftp.daum.net
 * updates: ftp.daum.net
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package vsftpd.i386 0:2.0.5-16.el5_5.1 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

=================================================================================
 Package              Arch          Version                  Repository      Size
=================================================================================
Updating:
 vsftpd               i386          2.0.5-16.el5_5.1         updates         141 k

Transaction Summary
=================================================================================
Install       0 Package(s)
Update        1 Package(s)
Remove        0 Package(s)

Total download size: 141 k
Downloading Packages:
vsftpd-2.0.5-16.el5_5.1.i386.rpm                               | 141 kB     00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
```

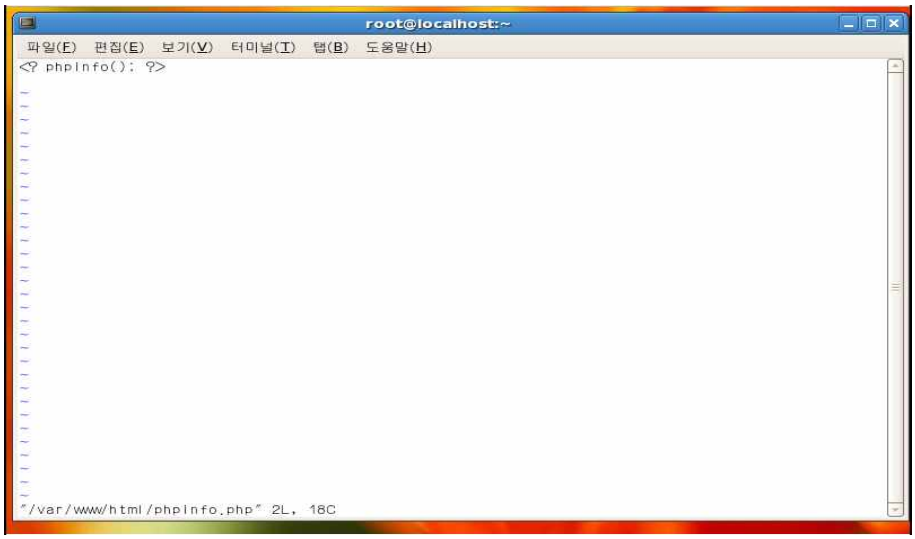
<소스 1-5>



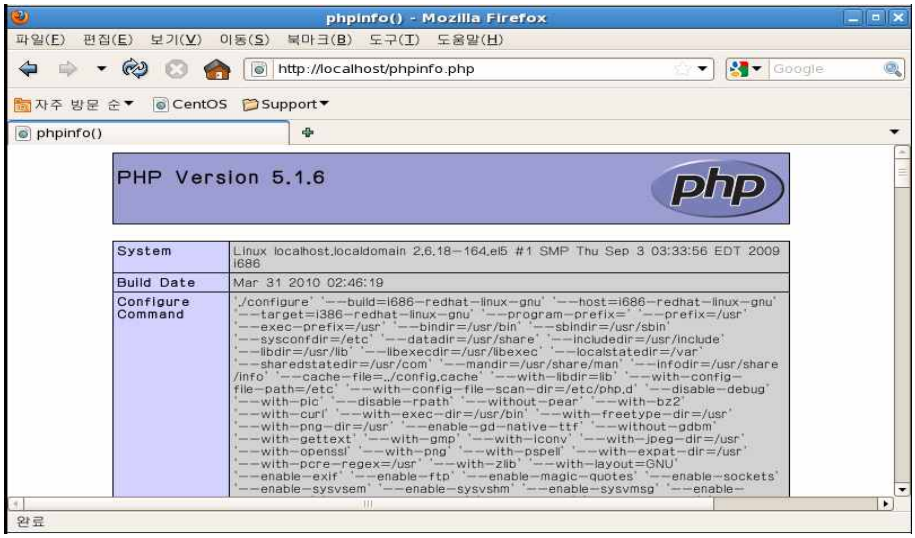
<소스 1-6>



<그림1> http://localhost



<그림1-1> <? phpinfo(); ?>



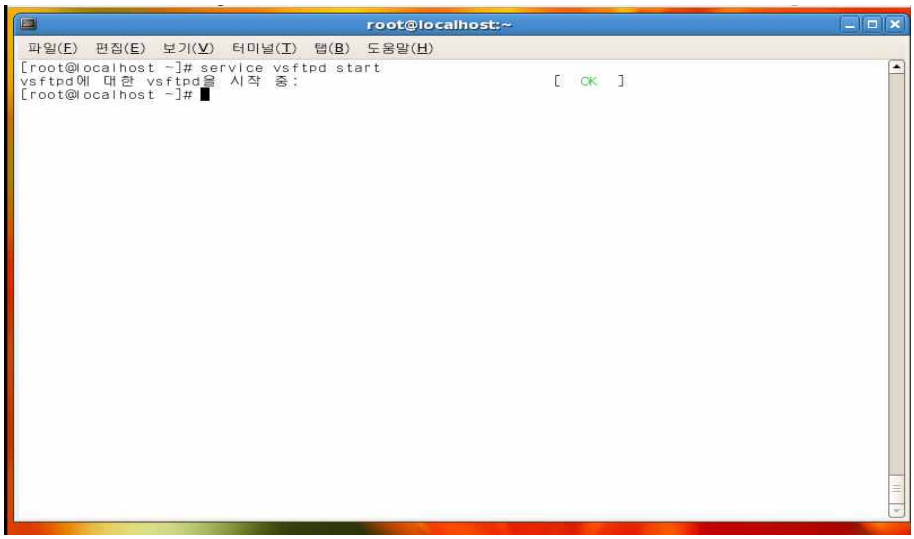
<그림1-2> http://localhost/phpinfo.php

```
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
o 4294967295
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:
/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h localhost.localdomain password 'new-password'

Alternatively you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default.  This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr ; /usr/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/bin/mysqlbug script!

The latest information about MySQL is available on the web at
http://www.mysql.com
Support MySQL by buying support/licenses at http://shop.mysql.com
                                                           [  OK  ]
MySQL (을)를 시작 중:                                      [  OK  ]
[root@localhost ~]#
```

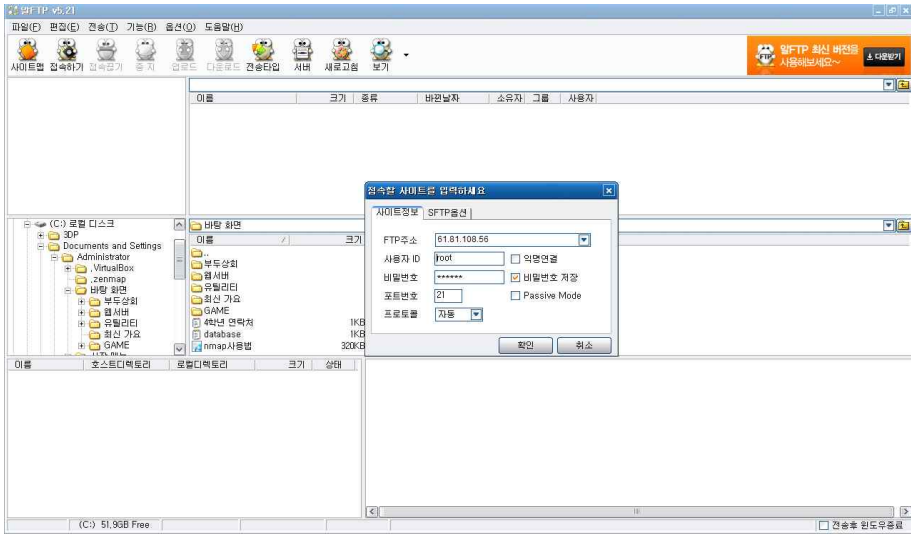
<소스 1-7> mysqld 데몬을 실행

```
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@localhost ~]# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 3
Server version: 5.0.77 Source distribution

Type 'help;' or '₩h' for help. Type '₩c' to clear the buffer.

mysql> set password for 'root'@'localhost' = password ('qwe123');
Query OK, 0 rows affected (0.00 sec)

mysql> create database zero;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
| zero               |
+--------------------+
4 rows in set (0.00 sec)

mysql> exit;
Bye
[root@localhost ~]#
```

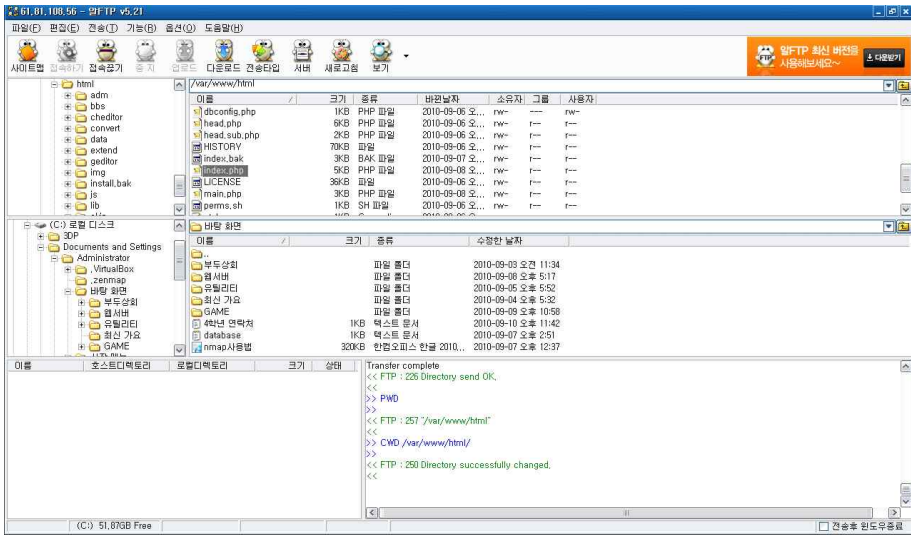
<소스 1-8> mysql root 비밀번호 설정 및 'zero' 테이블 생성.



<소스 1-9>

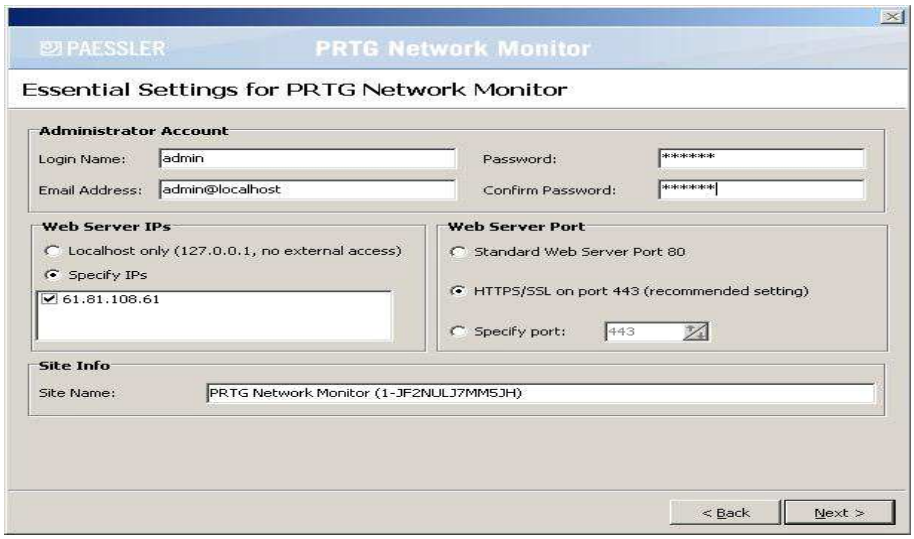


<그림1-3> FTP

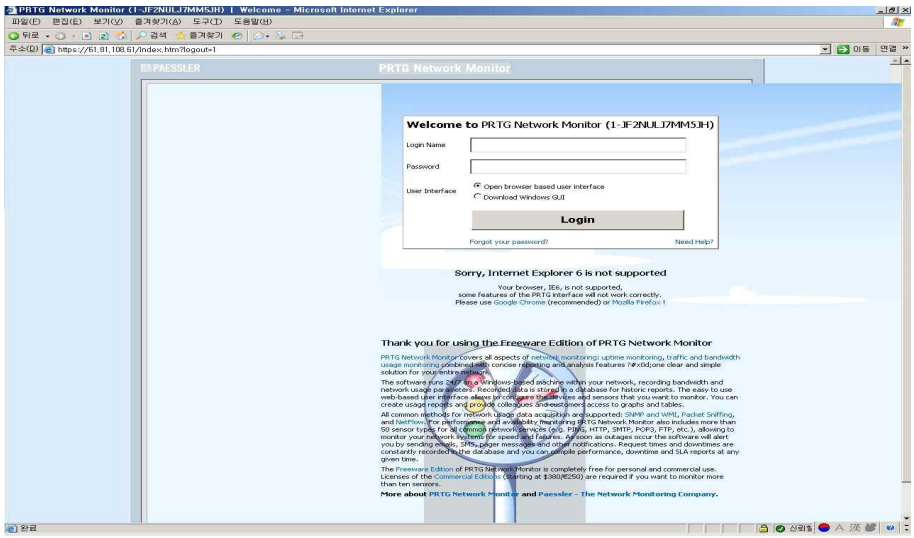


<그림1-4> FTP

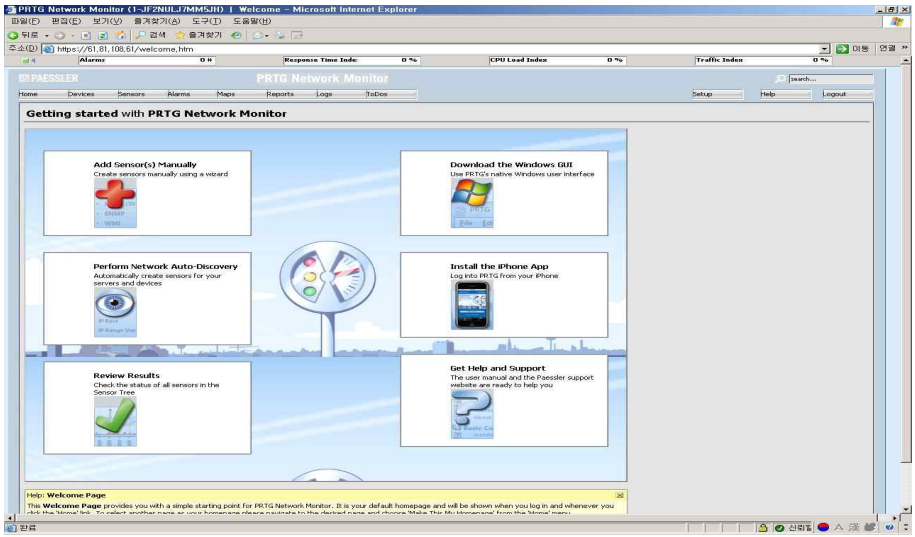
<그림 1-5> 구축된 Webserver (http://61.81.108.56)



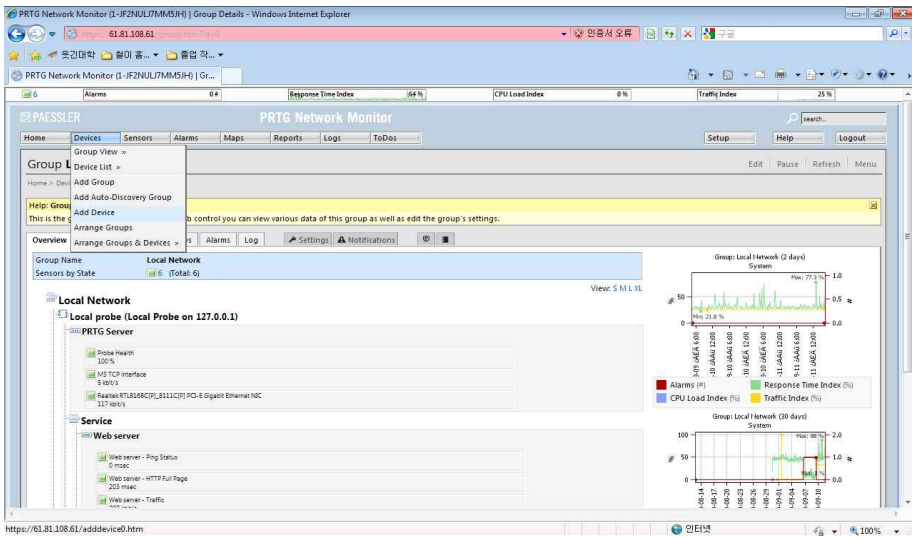
<그림 2-1> 관리자의 아이디와 비밀번호 메일을 설정 해준다.



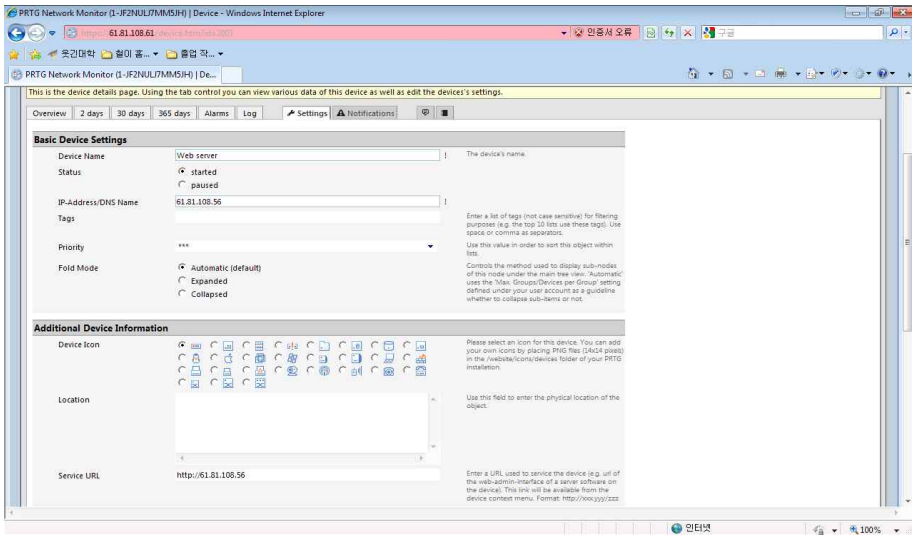
<그림 2-2> http://61.81.108.61 페이지로 접속해서 로그인을 한다.



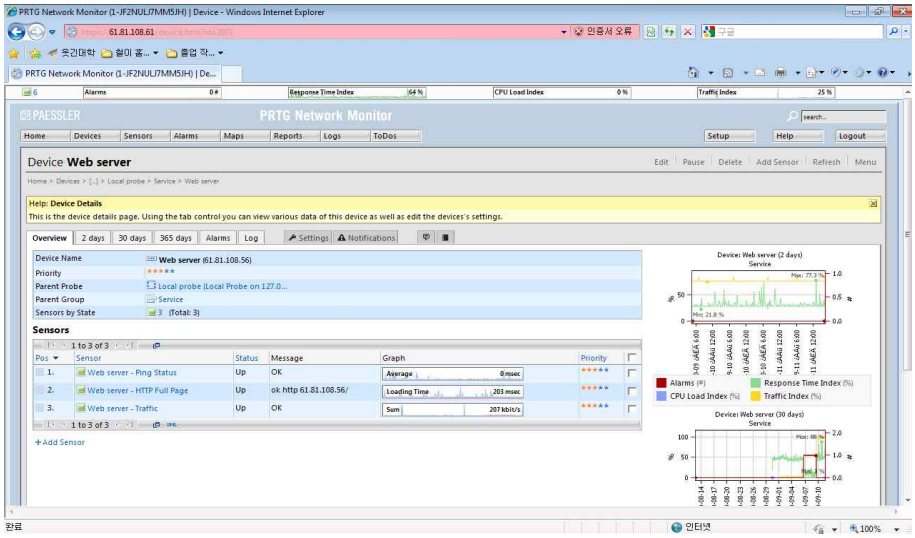
<그림 2-3> 기본 인터페이스 화면이다 이곳에서 Device를 추가한다.



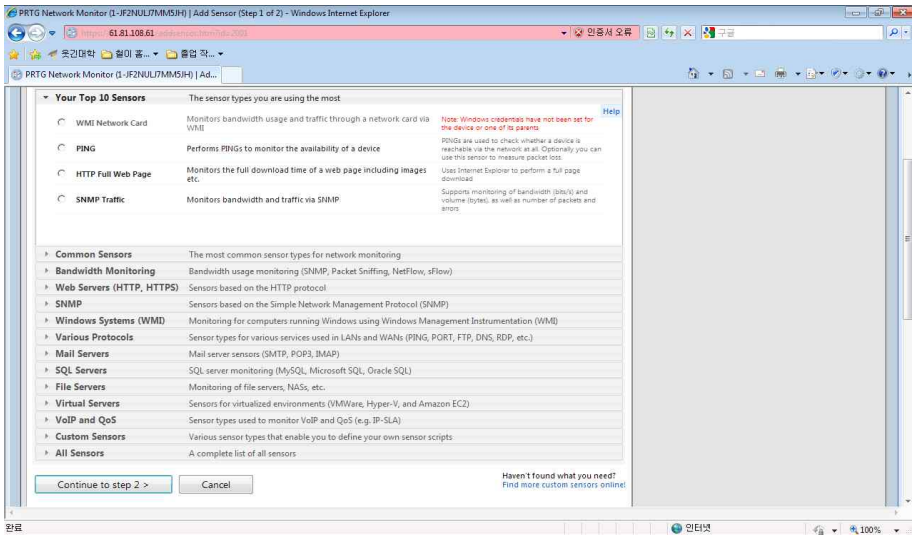
<그림 2-4> add device를 눌러서 장치를 추가 해준다.



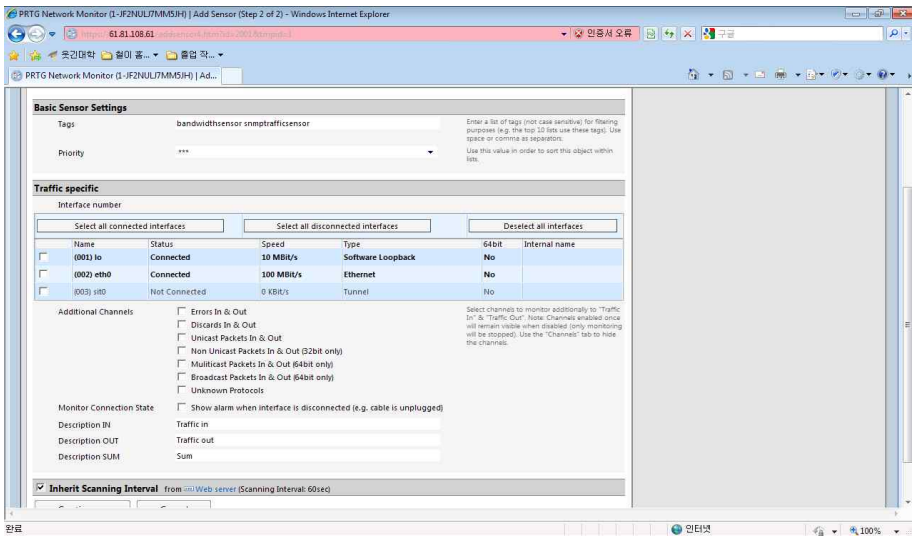
<그림 2-5> 서버의 상황에 맞게 IP주소를 추가 시켜준다.



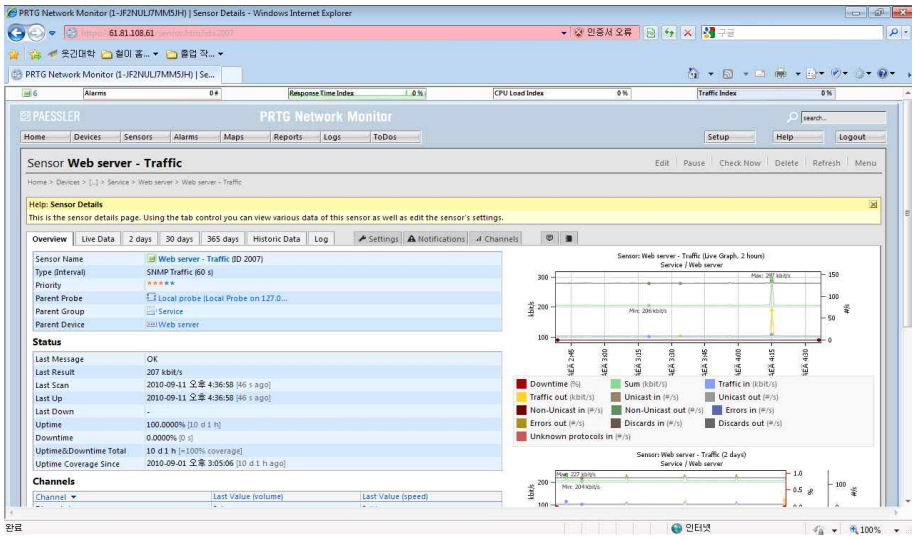
<그림 2-6> 제대로 설정이 되었다면 윗 그림과 같이 상태창이 보여지게 된다.



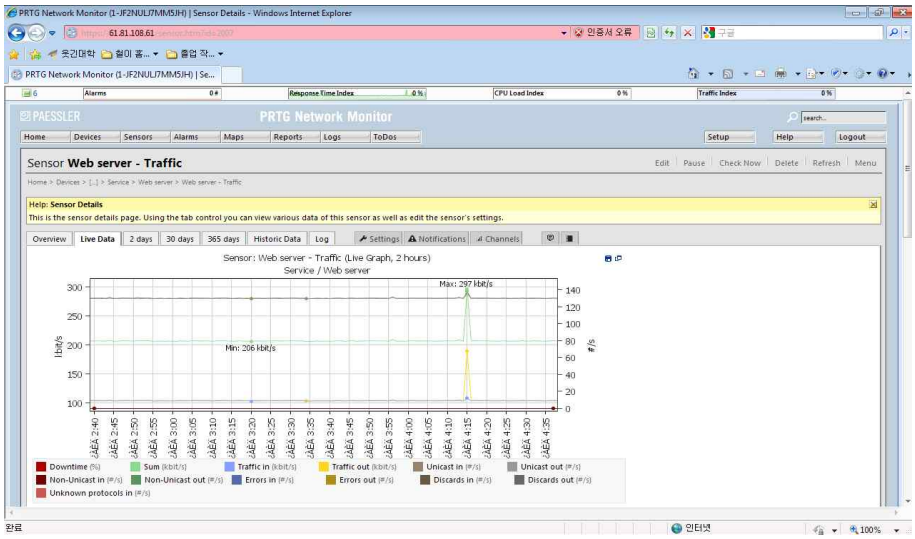
<그림 2-7> 센서를 추가해주는 부분으로써 자신이 필요한 센서를 추가시켜주는데 트래픽을 감시하기 위한 목적이므로 SNMP Traffic을 추가해준다.



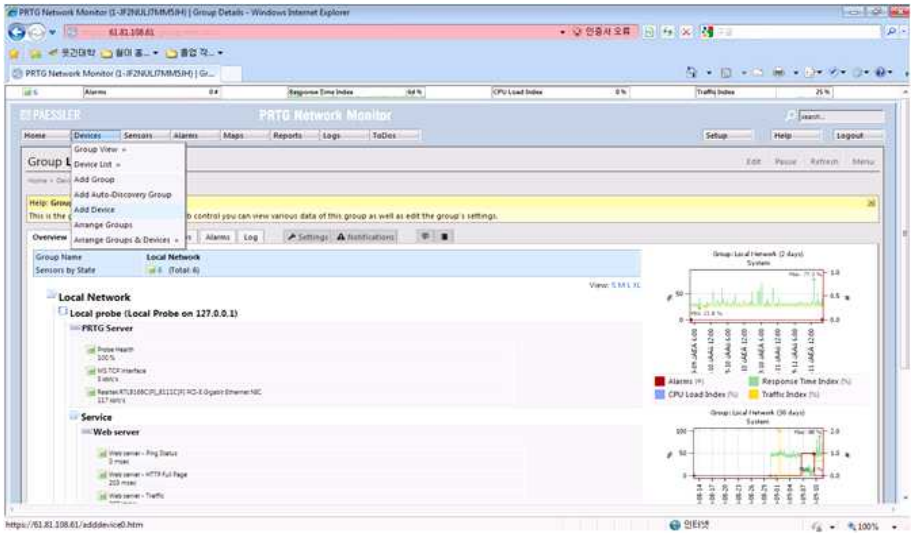
<그림 2-8> Traffic 추가 부분 설정 페이지로써 감시할 이더넷 장치와 감시 할 채널들을 설정 해준다.



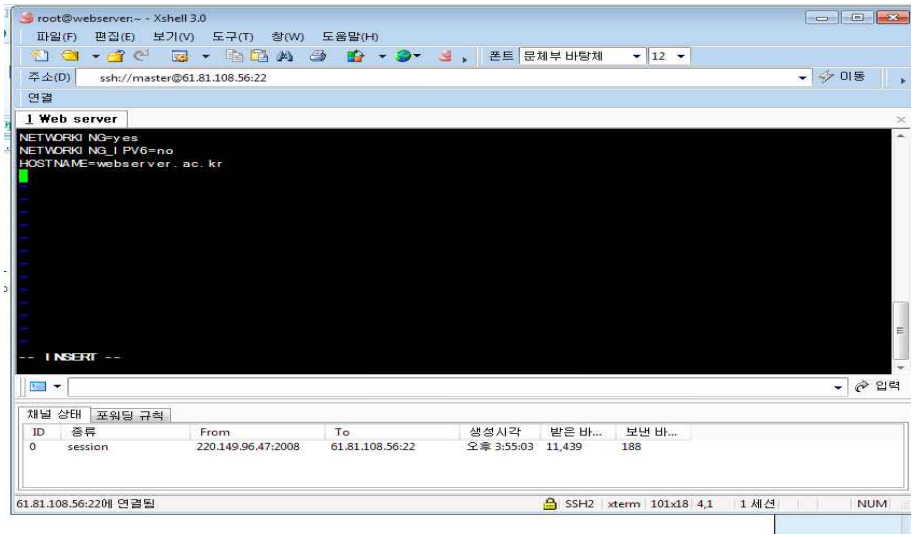
<그림 2-9> Traffic 설정이 제대로 이루어 지면 다음과 같은 페이지가 생성된다.



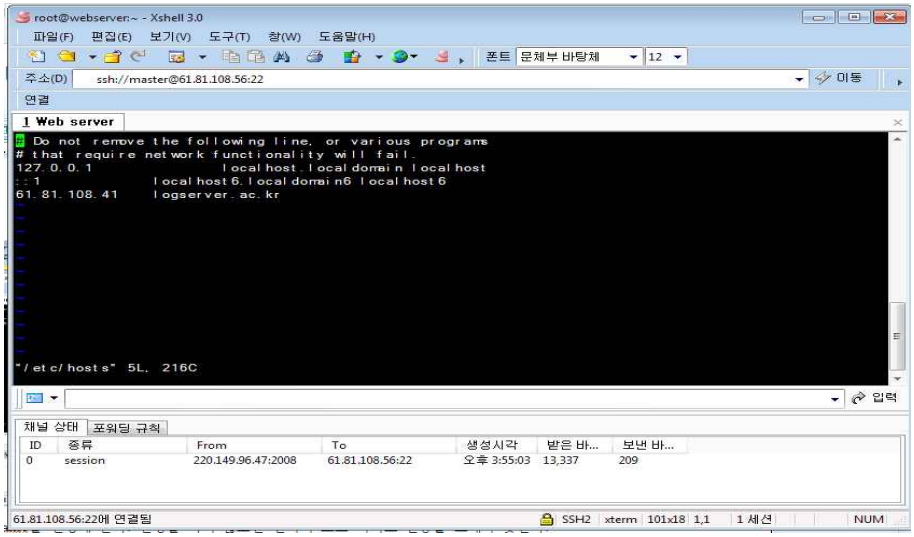
<그림 2-10> Traffic 감시 페이지이다. 초 단위로 Kbit/s 감시가 진행 중인 것을 확인할 수 있다.



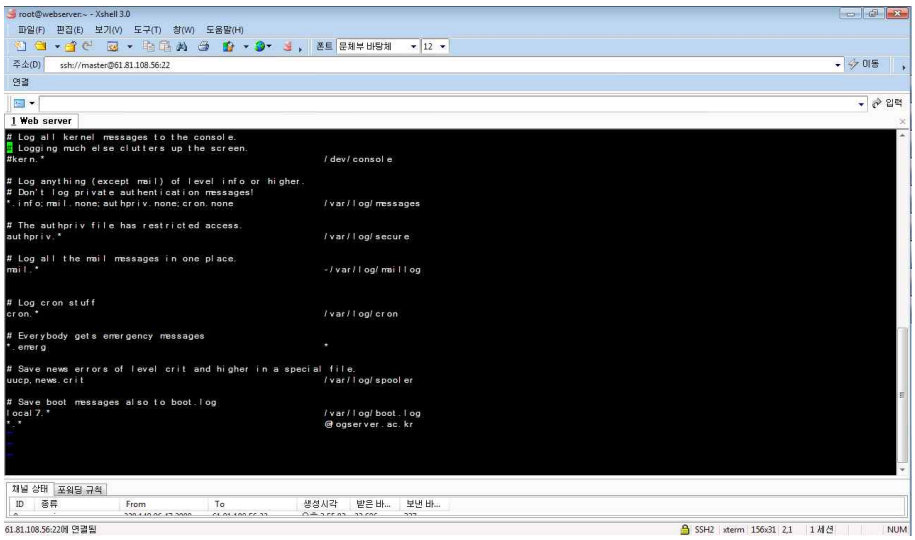
<그림 2-11> 구축된 PRTG server (http://61.81.108.61)



<그림 3-1> 먼저 hostname을 변경해 준다. 변경을 하지 않으면 원격지 로그 서버로 전송을 보내지 못한다.



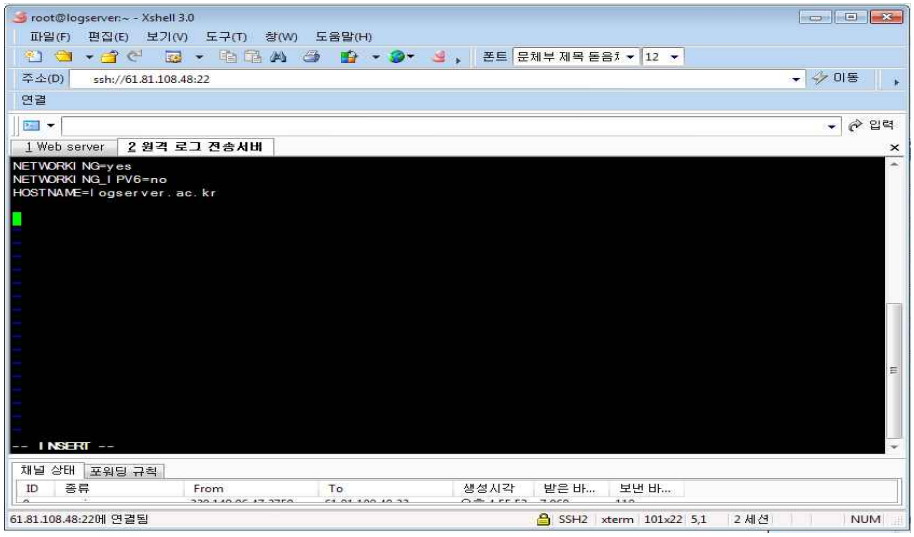
<그림 3-2> 원격지 로그 서버의 IP와 hostname을 입력해준다.



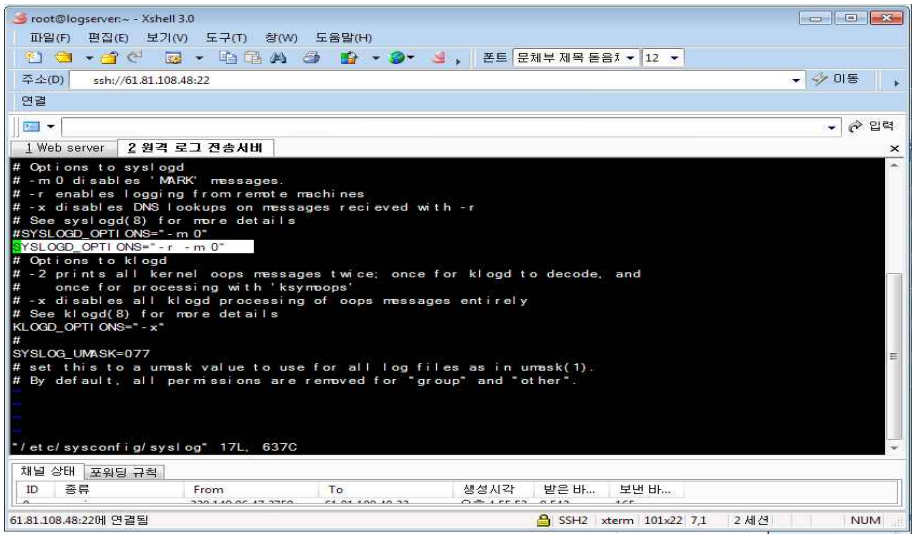
<그림 3-3> /etc/syslog.conf 파일을 편집기로 열어서 사용자가 원하는 로그를 원격지 로그로 설정해주면 되는데 모든 로그들을 전송하게 하기 위해서 제일 아래 줄에 다음과 같이 추가해주면 모든 로그를 원격지 로그 서버로 전송하게 된다.

.@logserver.ac.kr

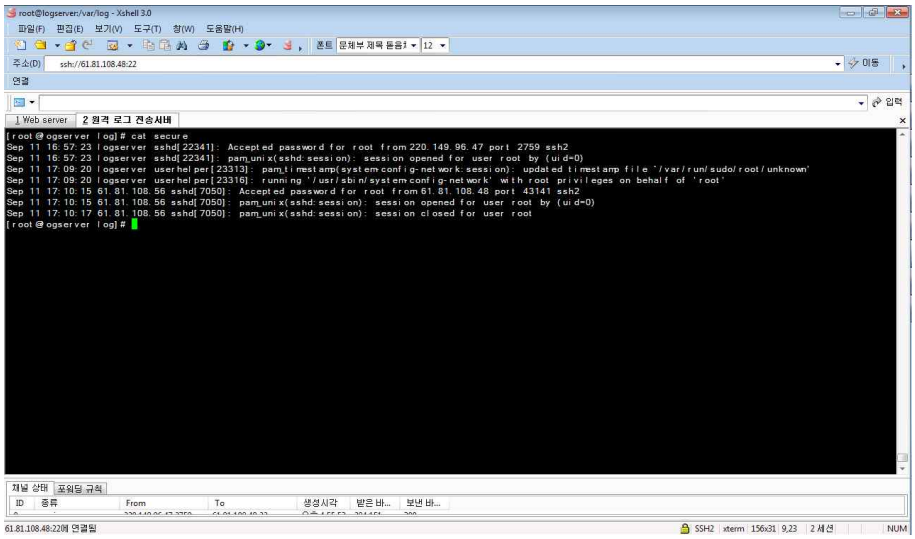
service syslogd restart 명령어로 로그 시스 데몬을 재시작 해준다.



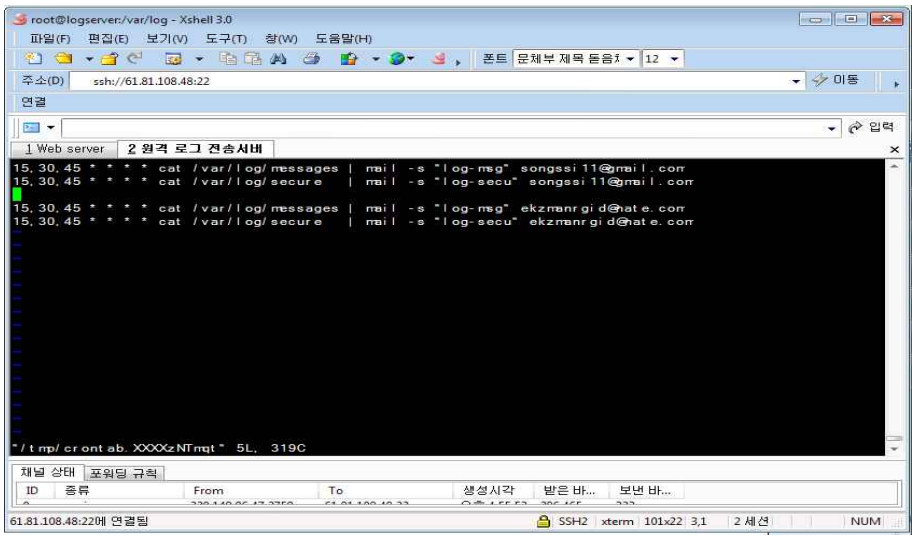
<그림 3-4> 원격지 로그 서버도 마찬가지로 hostname을 변경해준다.



<그림 3-5> 원격 로그의 허용을 하기 위해서 SYSLOGD_OPTIONS=" -m 0" 이라면 "-r -m 0"으로 수정 후 데몬을 재시작 한다.

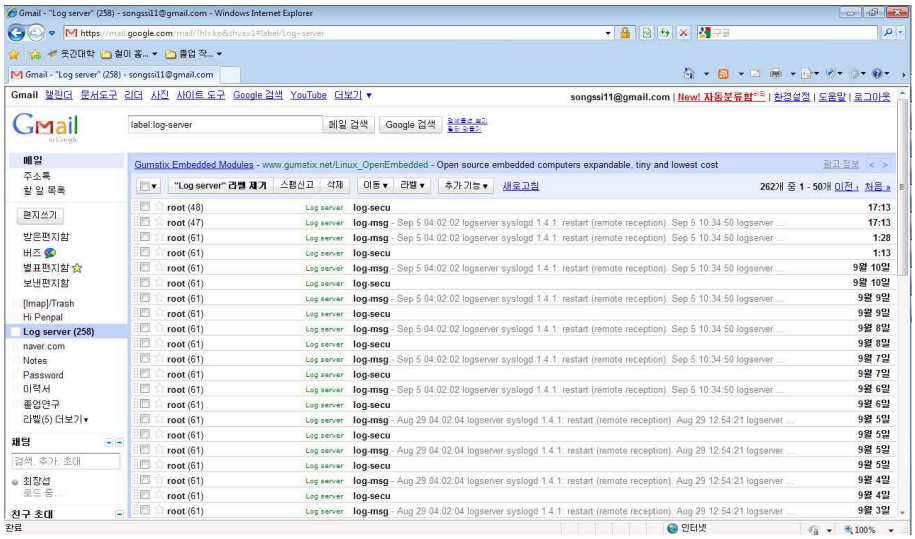


<그림 3-6> 61.81.108.56 (web server)에서 전송된 로그 파일을 전송 받았음을 확인할 수 있다.

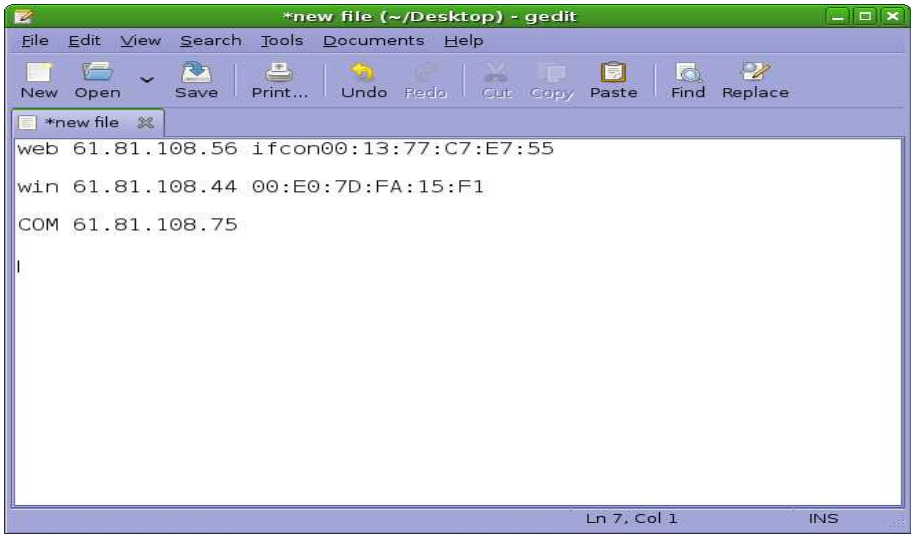


<그림 3-7> crontab -e 명령어 입력 후 편집 화면이 나오는데 여기서 관리자 메일로 15분마다 전송하게 하는 명령어를 입력 해두면 로그 파일은 15분마다 관리자 메일로 전송되게 된다.

15,30,40 * * * * cat /varlog/messages | mail -s "전송 시 파일이름" 관리자 메일

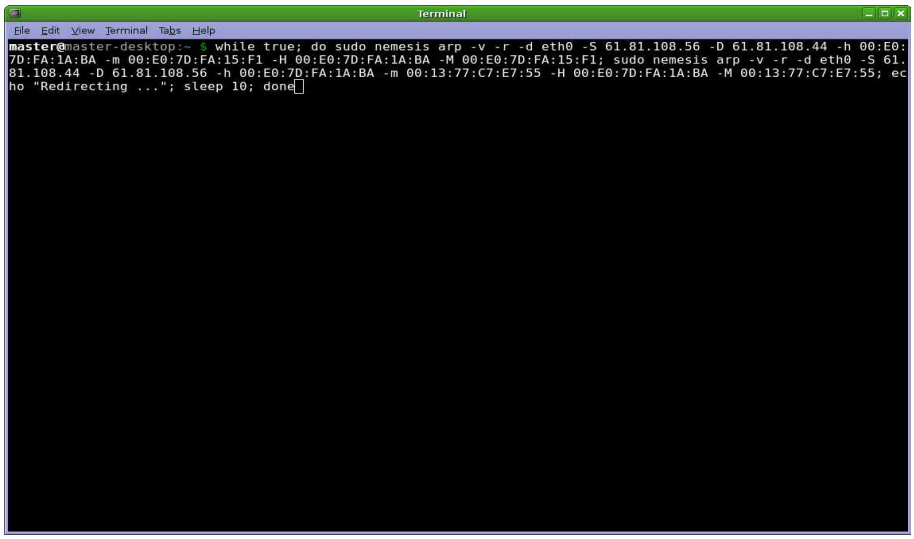


<그림 3-8> 15분 마다 관리자 메일로 전송된 로그 파일들

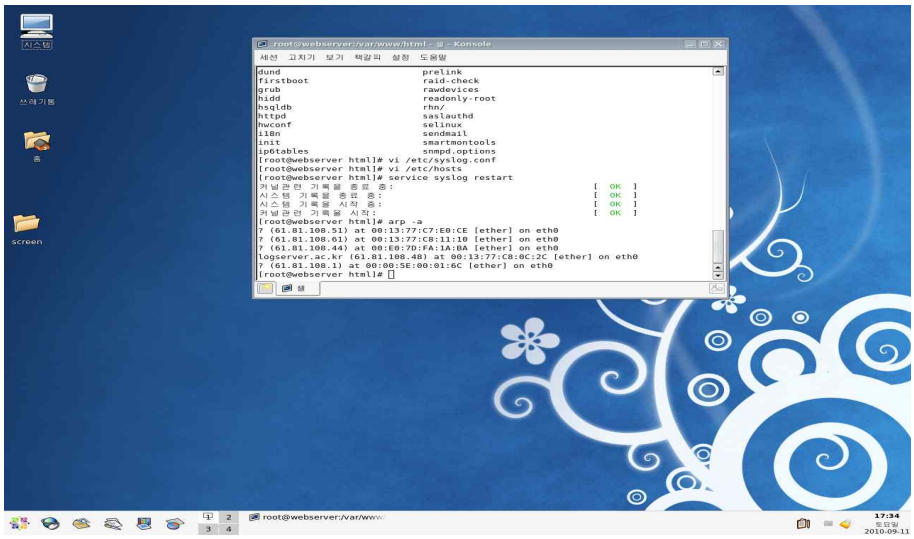


<그림 4-1> web server 와 win xp (공격자 pc) 서로 ping 을 보낸 후 apr -a 명령어로 mac 주소를 파악한다.

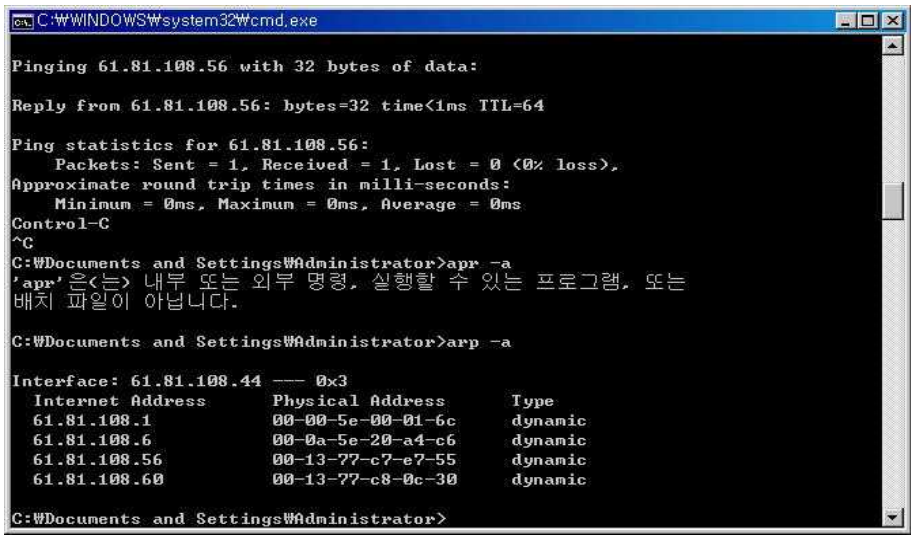
그리고 ip 포워드를 해주어야만 자신으로 들어오는 패킷을 상대방에게 재전송할 수 있게 되어 네트워크 가 작동한다.



<그림 4-2> nemeisis를 이용하여 arp 스푸핑을 걸고 web server 와 공격자 pc 의 MAC주소를 wireshark mac주소로 속여 보낸다.

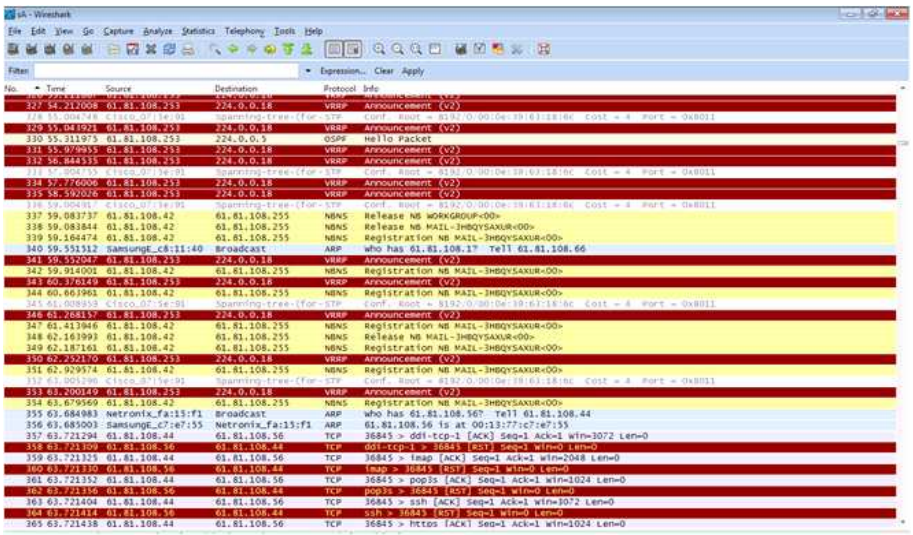


<그림 4-3>

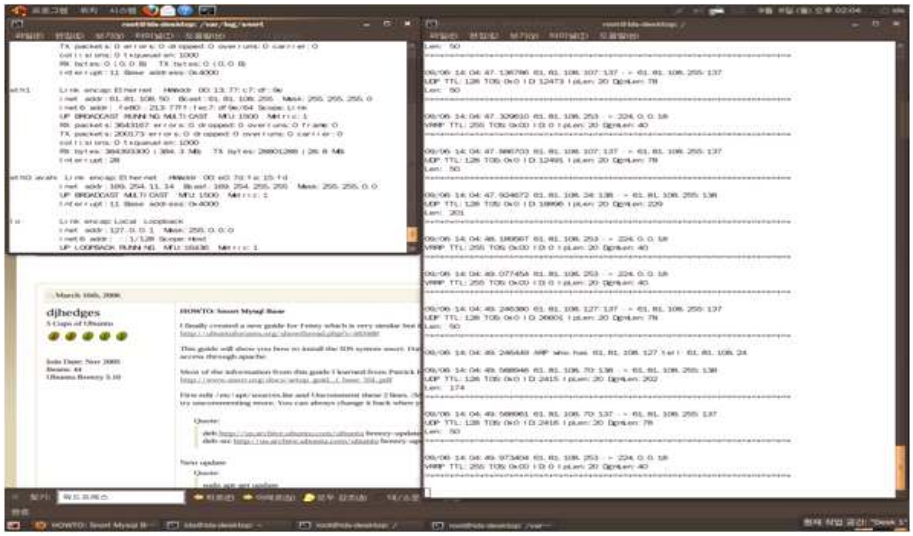


```
C:₩WINDOWS₩system32₩cmd.exe

Pinging 61.81.108.56 with 32 bytes of data:

Reply from 61.81.108.56: bytes=32 time<1ms TTL=64

Ping statistics for 61.81.108.56:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:₩Documents and Settings₩Administrator>apr -a
'apr'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.

C:₩Documents and Settings₩Administrator>arp -a

Interface: 61.81.108.44 --- 0x3
  Internet Address      Physical Address      Type
  61.81.108.1           00-00-5e-00-01-6c     dynamic
  61.81.108.6           00-0a-5e-20-a4-c6     dynamic
  61.81.108.56          00-13-77-c7-e7-55     dynamic
  61.81.108.60          00-13-77-c8-0c-30     dynamic

C:₩Documents and Settings₩Administrator>
```

<그림 4-4> arp가 제대로 걸리면 webserver 와 공격자pc 의 mac 주소는 wireshark 로 인식한다.



<그림 4-5> 구축된 Wireshark server



<그림 5> 구축된 Snort server

간지 넣어주세요

2010 졸업연구 결과보고서

Snort와 Nessus를 이용한 안전한 기업 환경 구축 및 운영

Building and Operation of Secure Corporate Environment Using Snort and Nessus

팀 명	보안시대
지도교수	이 병천 교수님
팀 원	서 우영 (4년)
	최 혁수 (4년)
	구 자연 (4년)
	김 면중 (4년)
	박 일환 (4년)
	최 미진 (4년)
Staff	최 장섭 (4년)
	김 선일 (4년)
	정 혜성 (3년)

2010. 10

중부대학교 정보보호학과

요 약 문

1. 연구제목

- Snort와 Nessus를 이용한 안전한 기업환경 구축 및 운영.
- Building and Operation of Secure Corporate Environment Using Snort and Nessus

2. 연구 목적 및 필요성

정보화 사회에서 정보란 대단히 중요하게 여겨진다. 이런 중요한 정보들이 쉽고 편리해진 인터넷 보급으로 인하여 외부로 유출될 확률이 높아졌다. 이를테면 기업의 기밀 정보가 외부로 유출이 되었을 시에 그 기업은 엄청난 경제적 타격은 물론 이미지에도 큰 타격을 입게 된다

이런 일을 미연에 방지하여 기업들의 원활한 활동을 지향할 필요성이 있다. Snort와 Nessus를 통한 네트워크를 구현하고 이에 따른 각 서버들을 구축하여 각종 외부 공격에 대한 패킷의 분석과 서버 혹은 사용자에게 어떠한 피해를 주는지에 대한 분석을 목적으로 한다.

3. 연구 내용

이에 Snort와 Nessus를 이용하여 전반적인 네트워크의 취약점 분석 및 패킷의 분석을 얻고 BASE로 화면상 보기 쉽게 전환하여 누구나가 자신이 사용하는 컴퓨터에 다른 누군가 몰래 침입을 했을 시에 알아 볼 수 있게 하기 위함이다.

4. 연구 결과

급격히 증가한 해킹으로 인한 정보유출에 우리는 외부공격이 어떠한 방법으로 접근을 하며 그에 대응할 수 있는 지가 궁금해짐에 방화벽 시스템 기술의 전반적인 이해와 개인 지식 향상과 각종 서버운영에 대한 실무능력과 취약점 분석 및 방지를 목적으로 연구를 하게 되었다. Snort 와 Nessus 를 통한 네트워크를 구현하고 이에 따른 각 서버들을 구축하여 각종 외부 공격에 대한 패킷의 분석과 서버 혹은 사용자에 어떠한 피해를 주는지에 대한 분석이 주목적이며, 또한 패킷 분석을 BASE를 통하여 한눈에 알아보기 쉽게 하여 외부 공격에 발 빠른 대처를 할 수 있는 방안을 목적으로 한다.

목 차

그림 목차

서 론

1. 연구 계획

1) 연구의 개요

최근 급격한 인터넷 보급의 확산으로 인하여 다양한 목적을 가진 악성코드나 해킹으로 인한 공격들로 개인 정보가 유출되거나 혹은 기업이 가지고 있던 기밀 정보들이 외부로 유출되는 경우가 많다. 그러나 인터넷을 사용하는 대부분의 사람들과 기업은 어떠한 방법, 경로로 정보가 유출되고 있는지에 대해 자세히 모르는 것이 현 실정이다.
이에 Snort와 Nessus를 이용하여 전반적인 네트워크의 취약점 분석 및 패킷의 분석을 얻고 BASE로 화면상 보기 쉽게 전환하여 누구나가 자신이 사용하는 컴퓨터에 다른 누군가 몰래 침입을 했을 시에 알아 볼 수 있게 하기 위함이다.

2) 연구의 필요성

정보화 사회에서 정보란 대단히 중요하게 여겨진다. 이를테면 기업의 기밀 정보가 외부로 유출이 되었을 시에 그 기업은 엄청난 경제적 타격은 물론 이미지에도 큰 타격을 입게 된다. 또 타 기업들에게 그 기밀 정보가 유포됨으로써 경쟁할 여력조차 잃게 된다.
이런 중요한 정보들이 쉽고 편리해진 인터넷 보급으로 인하여 외부로 유출될 확률이 높아졌다. 이런 일을 미연에 방지하여 기업들의 원활한 활동을 지향할 필요성이 있다.

3) 연구의 목적

보안 기술의 전반적인 이해와 개인 지식 향상과 각종 서버운영에 대한 실무능력과 취약점 분석 및 방지를 목적으로 한다. Snort 와 Nessus 를 통한 네트워크를 구현하고 이에 따른 각 서버들을 구축하여 각종 외부 공격에 대한 패킷의 분석과 서버 혹은 사용자에 어떠한 피해를 주는지에 대한 분석을 목적으로 한다. 또한 패킷 분석을 BASE를 통하여 한눈에 알아보기 쉽게 하여 외부 공격에 발빠른 대처를 할 수 있게 하기 위함이다.

4) Snort는 무엇인가

Snort는 “sniffer and more″라는 말에서 유래되었는데, 1998년 처음 공개되었을 때는 단지 2개의 파일 뿐이었으며 코드도 1,600줄밖에 되지 않는 단순한 패킷 스니퍼 프로그램이었다. 그러나 1999년 rule 기반의 분석 기능이 추가되어 이후 개발자 커뮤니티를 통한 계속 적인 기능 보안과 향상을 통해 지금과 같은 모습을 띄게 되었다.

5) Nessus는 무엇인가

Nessus는 자신의 네트워크의 취약점을 파악하는게 주목적이며 더불어 해결책도 제시해 줄 수 있게 현재 활발히 개발이 진행되고 있으며 linux계열의 또 다른 보안 프로젝트라고 할 수 있다. Nessus는 하루에도 새로운 해킹기법을 사용할 수 있게 해주는 스크립트가 새로 보강되는 살아 숨 쉬는 프로젝트라 할 수 있다.

6) BASE는 무엇인가

BASE는 Basic Analysis and Security Engine의 약자로서 ACID 프로젝트의 코드를 기반해서 만들어졌다. 이 응용프로그램은 Snort 침입 탐지 시스템에서 나오는 경고들을 질의하고 분석할 수 있게 해주는 웹 화면을 제공한다. Snort의 로그를 체계적이고 유연하게 보여질 수 있도록 구성된 BASE를 이용하면 분산 환경뿐만 아니라 단일한 Snort 센서에서의 탐지 역시 가능하고 관리자가 쉽게 알아볼 수 있게 하루 송신의 양, 하루에 몇 번을 접근하였는지 등을 그래픽 형식으로 보여준다.

본 론

2. 기반 기술

1) FTP Server

(1) FTP의 정의

FTP(File Transfer Protocol, 파일 전송 프로토콜)는 TCP/IP 프로토콜을 가지고 서버와 클라이언트 사이의 파일 전송을 하기 위한 프로토콜이다. 파일 전송 프로토콜은 TCP/IP 프로토콜 테이블의 응용 계층에 속하며, 예전에는 이 서비스가 파일을 전송하기 위해서 아주 널리 사용되어 왔으나, 웹 환경이 완전히 보급되면서 웹에서 FTP의 고유 기능인 파일전송을 편리하게 할 수 있게 되어서 예전보다 인기가 많이 떨어졌다. 하지만 아직도 파일 전송 자체를 위해서는 성능이 뛰어나며 일부 사이트에서도 계속 이 서비스가 제공되고 있다.

연결 시 수동 모드와 능동 모드가 있는데 그 안의 연결 종류에는 2가지가 있다.

· 능동 모드(포토 모드) : 서버가 자신의 데이터 포트인 20번 포트에서부터 클라이언트가 지정한 지점으로의 데이터 연결을 만든다. 클라이언트가 지정하는 포트는 주로 1023 보다 큰 번호가 매겨진 포트이다. 클라이언트가 방화벽, NAT(IP 마스킹) 등을 사용하는 환경일 때에 잘 동작하지 않을 수 있는데, 이때 수동 모드를 이용하면 된다.

· 수동 모드 : 클라이언트가 서버가 지정한 서버 포트로 연결할 수 있게 한다. 이런 때에는 보통 양쪽 포트 모두 1023 보다 큰 포트를 사용한다.

· 명령 연결 : 먼저 제어포트인 서버 21번 포트로 사용자 인증, 명령을 위한 연결이 만들어지고, 여기를 통해 클라이언트에서 지시하는 명령어가 전달된다.

· 데이터 전송용 연결 : 실제의 파일 전송은 필요할 때 새로운 연결이 만들어진다.

어느 모드에서도 2개의 연결을 만드는 점은 다르지 않다. 서버에 방화벽이 있는

경우, 데이터 연결에 어느 포트 번호를 사용할지를 설정해 방화벽과 문제는 없는지 확인해 보아야 한다. 수동 모드를 사용하고 있을 때에는 클라이언트의 방화벽은 신경 쓰지 않아도 된다.

(2) FTP의 필요성

서버 운영에 있어서 가장 중요한 요소는 보안문제이다. 기존의 ProFTP와 WU-FTP 는 보안 문제에 대한 보고가 빈번히 일어나 서버의 보안이 흔들린 경우가 많다. vsFTP는 보안부분을 특히 강조한 서버 데몬으로서 REDHAT, SUSE, OPEN-BSD에서 기본 FTP 데몬으로 채택하고 있으며 vsFTP를 매우 신뢰하고 있다. vsFTP에서 보안, 빠른 수행, 안정성을 주요 특징으로 소개하고 있고 그 성능도 여느 FTP서버 보다 탁월하다. 지원하는 대표적인 기능으로는 가상 IP 지원, 가상유저 지원, Standalon 과 inetd 지원, 강력한 사용자 설정, 전송 대역폭 조절기능, 환경설정파일을 IP별로 독립적 운영 지원, IP별 제한 기능 등이 있다. 또한 config 파일의 설정문법도 아주 간단해서 FTP 서버관리를 쉽게 할 수 있다. FTP를 이용하면 자신이 원하는 프로그램이나 각종 데이터를 무료나 저렴한 가격에 살 수 있다. 또 용량이 큰 파일도 빠르게 송수신할 수 있다. 파일을 송수신할 때에는 정당한 자격, 즉 원격 호스트 컴퓨터를 이용할 수 있는 사용자 ID와 패스워드(password)가 있어야 원하는 원격 호스트 컴퓨터에 접속할 수 있다. 그러나 인터넷상에는 패스워드가 없어도 접속할 수 있는 공개 FTP 호스트가 있다. 이러한 FTP 호스트를 Anonymous FTP라고 하는데 전 세계적으로 이러한 Anonymous FTP는 수천 개에 이른다. 사용자로 등록하지 않고서도 anonymous 라는 ID와 패스워드로 자신의 E-mail 주소를 설정하면 원격지 호스트에 접속하여 파일을 쉽게 송수신할 수 있다.

2) Web Server

(1) Web Server의 정의

Web이란 World, Wide, Web 의 준말로서 최근 인터넷에서 가장 인기를 끄는 정보연결서비스로 인터넷상의 다른 서비스와는 달리 문자 위주의 서비스가 아닌 문자·영상 음성 등 혼합된 멀티미디어 정보를 거미줄과 같은 통신망으로 세계로 연결시켜 주는 서비스이다.

초기의 Web 서버는 이미 준비해 놓은 파일을 송신하는 기능 밖에 갖지 않았으

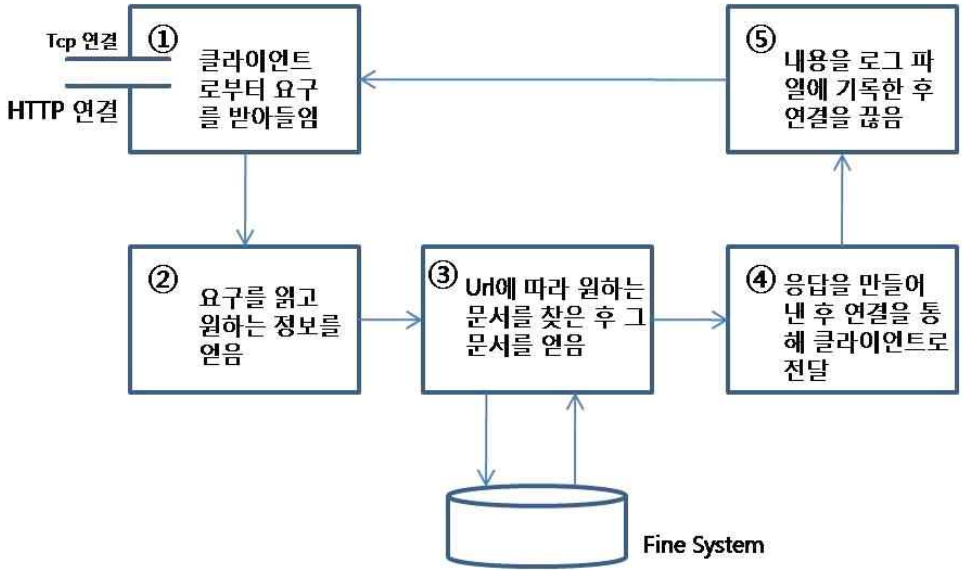
나, 최근에는 기능이 늘어서 요구에 따라 프로그램을 실행하여 결과를 클라이언트에 송신 하는 동적 페이지 생성 기능, 데이터베이스와 연계하여 트랜잭션 처리 기능 등을 가진 것도 등장하였다.

Web 서버 소프트웨어 중 가장 인기를 끌고 있는 것은 전 세계의 볼란티어 프로그래머가 공동 개발하고 있는 프리소프트웨어 Apache이다.

보편적인 웹 서버로는 아파치와 인터넷 인포메이션 서버 엔터프라이즈 서버 등이 있다. 이 웹 서버들은 전자 우편,파일 전송 규약(FTP) 파일의 내려받기 홈 페이지 구축, 전자 상거래 등에 필요한 인터넷 및 인트라넷과 관련된 프로그램들의 패키지의 일부로 나온다.

따라서, Web 서버란 인터넷이라는 네트워크에서 개인 사용자들이 업체의 컴퓨터에 접속하여 홈페이지 방문 혹은 기타 서비스를 제공하기 위하여 인터넷 네트워크에 구축해 놓는 컴퓨터 시스템이다.

· Web Server의 동작 원리



<그림 1-1 Web 서버의 동작 원리>

(2) Web Server의 필요성

시대가 빠르게 변화함에 따라 세상을 살면서 많은 정보를 필요로 하게 되었다. 이에 기업체들은 사용자의 요구를 충족 시키기 위해 웹을 통해서 언제 어디서든 필요한 정보를 웹서버를 통해 제공하고 있으며 그 대표적인 예로 온라인 쇼핑몰, N사의 지식인, 지도 서비스 등이 우리 주변 생활에 익숙해져있다.

3) Nessus

(1) Nessus의 정의

Nessus는 자신의 네트워크의 취약점을 파악하는게 주목적이며 더불어 해결책도 제시해 줄 수 있게 현재 활발히 개발이 진행되고 있으며 linux계열의 또 다른 보안 프로젝트라고 할 수 있다. 또한 서버는 Nessus 데몬과 플러그인이 설치되고 클라이언트에는 Nessus 서버를 통해 취약점을 발견할 수 있는 GUI 또는 명령어가 설치가 된다. 서버 구축 시 유닉스 계열만이 가능하지만 클라이언트는 유닉스 계열은 물론 윈도우, MAC등 여러 OS에서 사용 할 수 있다.

(2) Nessus의 필요성

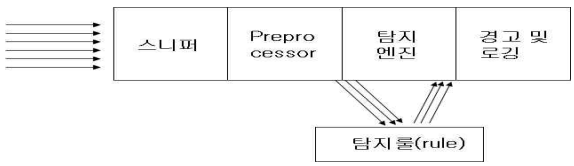
Nessus는 바로 지금 진행 중인 최신 해킹기법을 사용해서 자신의 시스템을 검사한다는 점이 다르다. Nessus는 하루에도 새로운 해킹기법을 사용할 수 있게 해주는 스크립트가 새로 보강되는 살아 숨 쉬는 프로젝트라 할 수 있다.

4) Snort

(1) Snort의 정의

Snort는 “sniffer and more″라는 말에서 유래되었는데, 1998년 처음 공개되었을 때는 단지 2개의 파일 뿐이었으며 코드도 1,600줄밖에 되지 않는 단순한 패킷 스니퍼 프로그램이었다. 그러나 1999년 rule 기반의 분석 기능이 추가되어 이후 개발자 커뮤니티를 통한 계속 적인 기능 보안과 향상을 통해 지금과 같은 모습을 띄게 되었다.

· Snort의 구조



<그림 1-55 Snort의 구조>

먼저 스니퍼를 통해 모든 패킷을 수집한 후, 수집된 데이터는 바로 탐지 엔진을

거치지 않고 그 전에 전처리기라 불리는 Preprocessor를 통해 보다 효율적인 공격 탐지를 위해 HTTP 인코딩 plug-in이나 포트스캔 plug-in등 몇 가지 plug-in을 먼저 거치면서 매칭이 되는지 확인하게 된다. Preprocessor 를 통과한 패킷은 snort IDS의 핵심이라 할 수 있는 탐지엔진을 거치면서 탐지 룰과 매칭되는지 확인하게 된다.

(2) Snort의 필요성

Snort는 패킷 스니퍼(sniffer)와 패킷 로거(logger) 그리고 Network IDS 기능이 있다. 패킷 스니퍼는 네트워크의 패킷을 읽어 보여주는 기능을 하고 패킷 로거는 모니터링 한 패킷을 저장하고 로그에 남기는 기능, 그리고 Network IDS는 네트워크 트래픽을 분석하여 공격을 탐지하는 기능으로 Bufferoverflow나 port scan, IP scan 등 대부분의 공격을 탐지할 수 있어 보안 및 크래커의 침입 탐지에 유용하다.

5) BASE

(1) BASE의 정의

BASE는 Basic Analysis and Security Engine의 약자로서 ACID 프로젝트의 코드를 기반해서 만들어졌다. 이 응용프로그램은 Snort 침입 탐지 시스템에서 나오는 경고들을 질의하고 분석할 수 있게 해주는 웹 화면을 제공한다. Snort의 로그를 체계적이고 유연하게 보여질 수 있도록 구성된 BASE를 이용하면 분산 환경뿐만 아니라 단일한 Snort 센서에서의 탐지 역시 가능하고 관리자가 쉽게 알아볼 수 있게 하루 송신의 양, 하루에 몇 번을 접근하였는지 등을 그래픽 형식으로 보여준다.

(2) BASE의 필요성

Snort의 분석 화면이 전부 문자열로 표기되는 관계로 한눈에 알아보기 쉽게 하기 위하여 BASE를 이용하여 그래픽화 하여 사용자가 더욱 쉽게 파악 할 수 있다. 또한 상업적 검색엔진에서 놓치거나 많은 양의 공격을 놓친 자세한 점을 밝혀낸다.

3. 구축방법

1) FTP의 구축방법

```
wildhb@localhost:/
파일(F)  편집(E)  보기(V)  터미널(T)  도움말(H)
[root@localhost /]# yum -y install vsftpd
Loaded plugins: refresh-packagekit
updates/metalink                                          | 9.6 kB     00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package vsftpd.i586 0:2.1.2-2.fc11 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package        Arch        Version                Repository          Size
================================================================================
Installing:
 vsftpd         i586        2.1.2-2.fc11           updates             159 k

Transaction Summary
================================================================================
Install       1 Package(s)
Update        0 Package(s)
Remove        0 Package(s)

Total download size: 159 k
```

<그림 3-1 FTP 설치>

· “yum -y install vsftpd” 명령어를 이용하여 강제 설치를 한다.

```
wildhb@localhost:/
파일(F)  편집(E)  보기(V)  터미널(T)  도움말(H)
================================================================================
Install       1 Package(s)
Update        0 Package(s)
Remove        0 Package(s)

Total download size: 159 k
Downloading Packages:
vsftpd-2.1.2-2.fc11.i586.rpm                              | 159 kB     00:01
경고: rpmts_HdrFromFdno: Header V3 RSA/SHA256 signature: NOKEY, key ID d22e77f2
updates/gpgkey                                            | 1.6 kB     00:00
Importing GPG key 0xD22E77F2 "Fedora (11) <fedora@fedoraproject.org>" from /etc/
pki/rpm-gpg/RPM-GPG-KEY-fedora-i386
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : vsftpd                                                    1/1

Installed:
  vsftpd.i586 0:2.1.2-2.fc11

Complete!
[root@localhost /]#
```

<그림 3-2 FTP 설치확인>

· Installed : vsftpd.i586 0: 2. 1. 2-2. fc11 ’란 메시지를 통해 설치가 되었음을 확인 할 수 있다.

```
[root@localhost pub]# cp /root/install* .
[root@localhost pub]# ls
install.log  install.log.syslog
[root@localhost pub]# service vsftpd start
vsftpd에 대한 vsftpd을 시작합니다:                          [  OK  ]
[root@localhost pub]#
```

<그림 3-3 vsFTPd 시작>

· ‘service vsftpd start’ - 데몬 명령을 사용하여 vsftpd 서비스를 시작한다.

· 'vsftpd에 대한 vsftpd 을 시작합니다 : [OK] - 메시지를 통해 서비스 시작을 확인할 수 있다.

```
wildhb@localhost:/var/ftp/pub
파일(F)  편집(E)  보기(V)  터미널(T)  도움말(H)
# Allow anonymous FTP? (Beware – allowed by default if you comment this out).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
-- INSERT --
```

<그림 3-4 vsFTPd 설정>

· vsftpd 설정

/etc/vsftpd 에 위치한 vsftpd.conf 파일을 vi 편집기를 이용, 아래의 부분을 주석을 제거하여 수정한다.

anonymous_enable=Yes ⇒ anonymous_enable=Yes

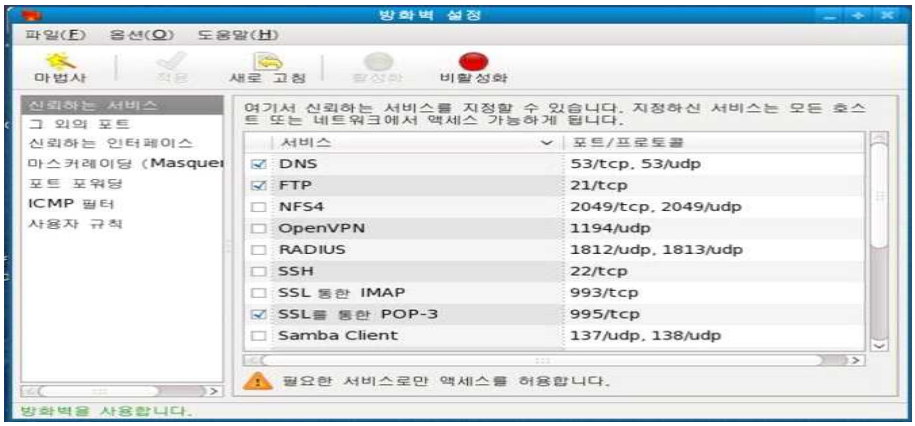
→ 익명 사용자가 접속을 허가할지 설정.

local_enble=Yes ⇒ local_enble=Yes

→ 로컬 사용자의 접속 허가를 설정.

write_enable=Yes ⇒ write_enable=Yes

→ 로컬 사용자가 저장, 삭제, 디렉토리 생성 등의 명령 수행의 허가를 설정. (익명 사용자는 해당 없음)



<그림 3-5 리눅스 방화벽 설정>

· 방화벽 설정

외부접속 허용을 위해 방화벽 설정을 한다. 명령어 'system-config-firewall' 또는 '시스템-관리-방화벽'을 이용하여 FTP 서비스를 실행시킬 수 있게 FTP에 체크 후 적용한다.

· 'server vsftpd restart' - 데몬 명령을 사용하여 vsftpd 서비스 재시작한다.

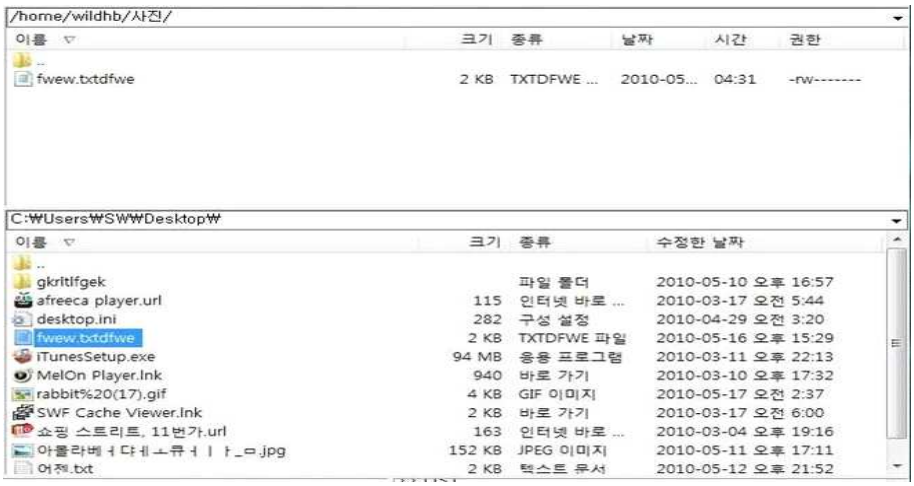
<그림 3-6 FTP 접속>

· 구축된 FTP 서버의 작동이 원활한지 접속 테스트를 한다. FTP 클라이언트에 FTP서버 IP를 입력 후 접속 확인한다. (익명 연결, 계정 연결 둘 다 확인해보는 것이 좋다.) 혹여 접속이 원활하지 않다면, 인터넷 연결 문제일 가능성이 있으니 인터넷 연결 확인을 꼭 해준다.

<그림 3-7 FTP 연결모습>

· FTP 서버에 연결된 모습.

익명으로 연결하게 되면 보통 /home/ 디렉토리로 접속이 된다. 계정을 만들어 계정으로 연결하게 되면 /home/'계정ID' 로 접속하게 된다.



<그림 3-8 FTP 업로드>

· FTP서버에 업로드 확인 테스트.

위에 설치 작업 중 /etc/vsftpd 에 위치한 vsftpd.conf 파일을 수정할 때 업로드를 가능하도록 수정을 했기에 업로드가 되는지 확인을 해준다.

2) Web의 구축방법

여러 가지의 구축방법 중에서도 APM을 이용한 구축방법을 사용하였다.

<그림 3-9 Mysql 설치>

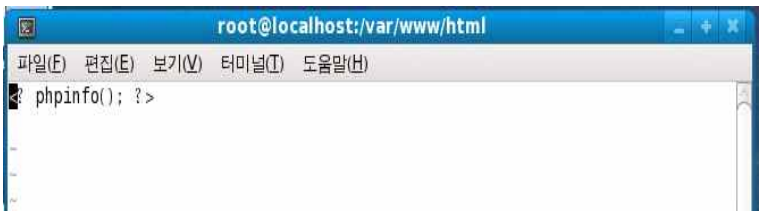
· yum명령어를 이용 하여 mysql를 설치 mysql을 설치하기 위하여 위의 이미지에 있는 'yum -y install mysql-server'를 사용하여 mysql을 설치한다.
· 설치 후 사용하기 위해 mysql을 접속하여 root 비밀번호를 사용자가 사용할 수 있게 설정한다.
· 명령어는 set password for '계정' @ 'localhost' = password('비밀번호')로 설정한다.

<그림 3-10 Datadases 생성 확인>

· 비밀번호 설정 후 mysql에 접속하여 사용할 databases를 생성한다. 생성 명령어는 'create database data;'로 사용하여 data를 생성한다.
· 제대로 설정되었는지 확인하기 위해 show databaes;을 이용하여 확인한다.
· databases 가 생성되었음을 확인할 수 있다.

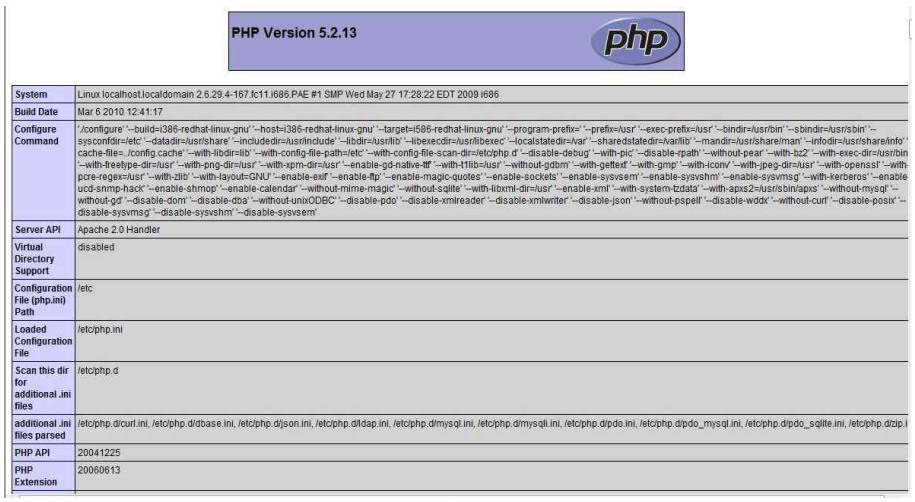
<그림 3-11 PHP 설치>

· yum명령어를 이용 하여 php를 설치 - php를 설치하기 위해서 위의 이미지에 있는 명령어 'yum -y install php* server' 를 사용하여 php를 설치한다.



<그림 3-12 PHPinfo 생성>

· php 설치 후 확인을 위해 var/www/html에서 vi편집기를 이용하여 다음과 같은 파일을 생성한다. 생성 파일명은 'phpinfo.php'이다.
· 위 이미지의 내용을 입력한 후에 저장하고 빠져 나온다.



PHP Version 5.2.13

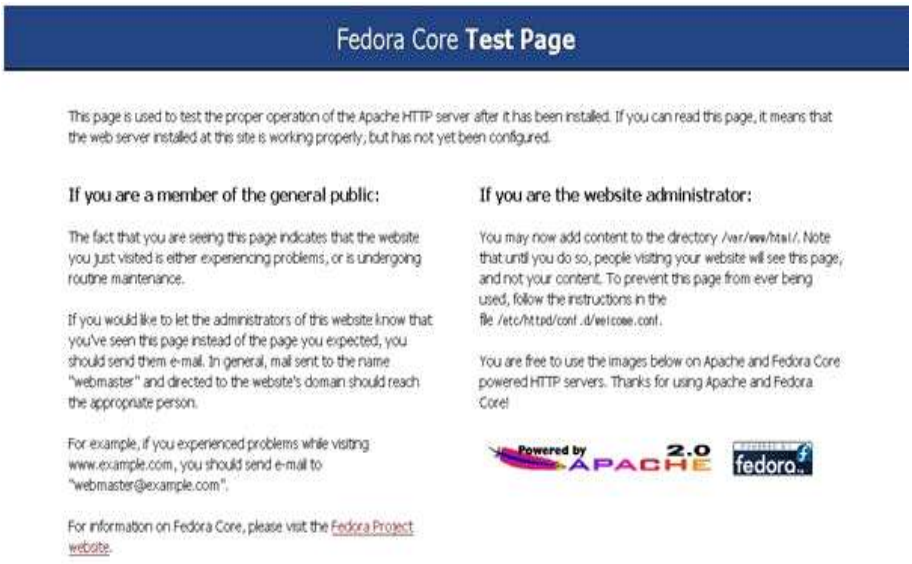
System	Linux localhost.localdomain 2.6.29.4-167.fc11.i686.PAE #1 SMP Wed May 27 17:28:22 EDT 2009 i686
Build Date	Mar 6 2010 12:41:17
Configure Command	'./configure' '--build=i386-redhat-linux-gnu' '--host=i386-redhat-linux-gnu' '--target=i586-redhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/var/lib' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=../config.cache' '--with-libdir=lib' '--with-config-file-path=/etc' '--with-config-file-scan-dir=/etc/php.d' '--disable-debug' '--with-pic' '--disable-rpath' '--without-pear' '--with-bz2' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--with-xpm-dir=/usr' '--enable-gd-native-ttf' '--with-t1lib=/usr' '--without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-pcre-regex=/usr' '--with-zlib' '--with-layout=GNU' '--enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-sockets' '--enable-sysvsem' '--enable-sysvshm' '--enable-sysvmsg' '--with-kerberos' '--enable-ucd-snmp-hack' '--enable-shmop' '--enable-calendar' '--without-mime-magic' '--without-sqlite' '--with-libxml-dir=/usr' '--enable-xml' '--with-system-tzdata' '--with-apxs2=/usr/sbin/apxs' '--without-mysql' '--without-gd' '--disable-dom' '--disable-dba' '--without-unixODBC' '--disable-pdo' '--disable-xmlreader' '--disable-xmlwriter' '--disable-json' '--without-pspell' '--disable-wddx' '--without-curl' '--disable-posix' '--disable-sysvmsg' '--disable-sysvshm' '--disable-sysvsem'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/dbase.ini, /etc/php.d/json.ini, /etc/php.d/ldap.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/zip.i
PHP API	20041225
PHP Extension	20060613

<그림 3-13 PHP 설치 후 페이지>

· 웹브라우저 주소창에 'http://localhost/phpinfo.php'를 입력하면 위와 같은 페이지가 나온다.

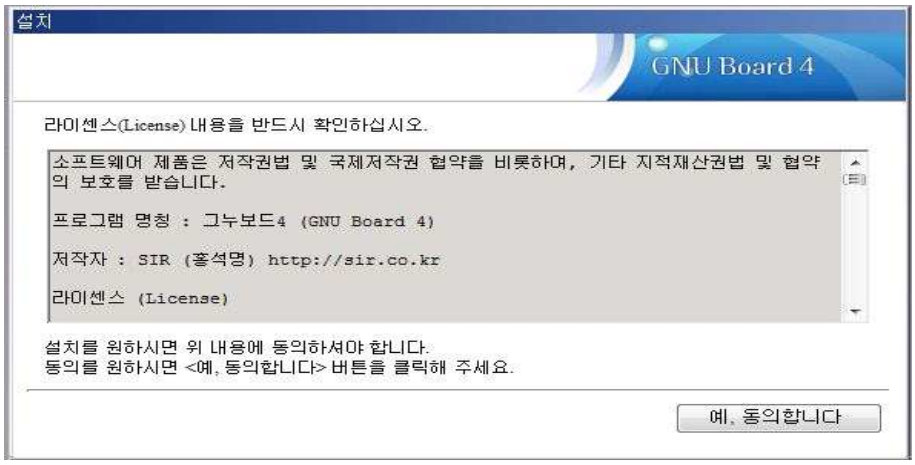
<그림 3-14 HTTPD 설치>

· yum명령어를 이용 하여 apache 를 설치 - apache를 설치하기위해 위에 이미지에 나와 있는 명령어 'yum -y install httpd server'를 이용하여 설치를 한다.



<그림 3-15 HTTPD 설치 후 페이지>

· 설치 후 'http://localhost' 웹 브라이저에 입력하면 위와 같은 페이지가 나온다.

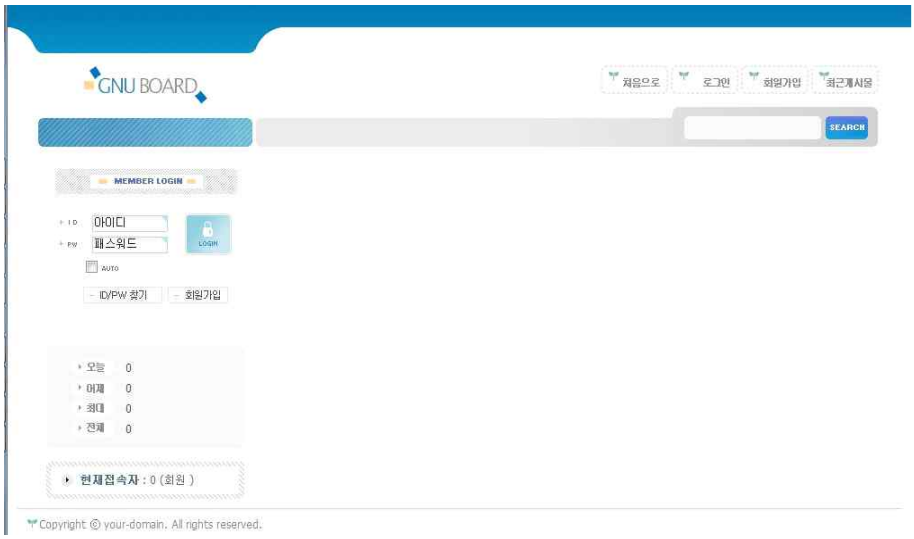


<그림 3-16 그누보드 설치 1>

· 그누보드를 이용한 설치 - 그누보드 홈페이지 'http://sir.co.kr/' 에서 최신 버전을 다운로드하여 '/var/www/html'에 압축을 풀어준다.
· 그 후 'http://localhost' 웹브라우저에 입력하여 설치를 시작한다.

<그림 3-17 그누보드 설치 2>

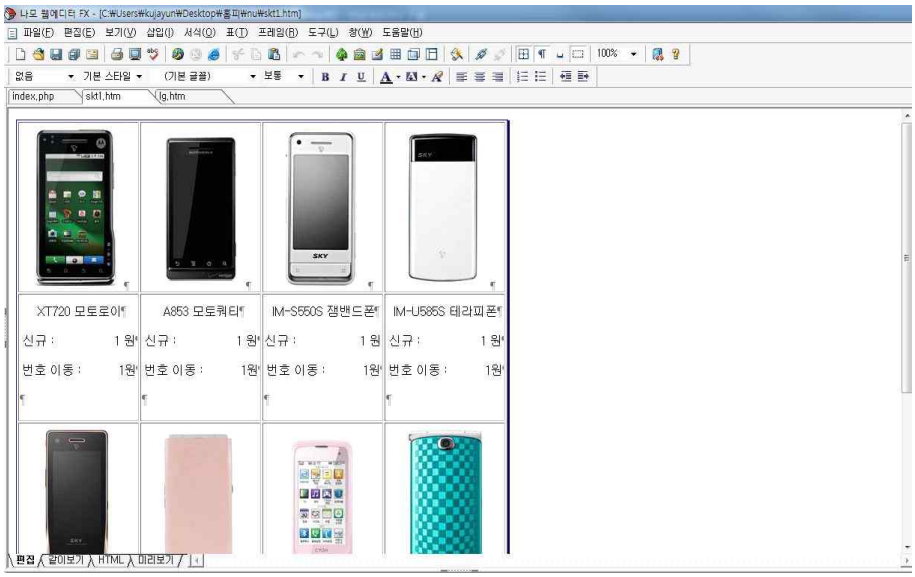
· User에는 root를 입력하고 Password는 root 비밀번호를 입력, DB는 mysql에서 만든 databases Data를 입력한다.
· 최고 관리자 password는 자신만이 사용 할 password를 입력 한 후에 설치를 계속한다.



<그림 3-18 그누보스 설치 후 페이지>

· 모든 설치가 완료되면 ‘http://61.81.108.51’로 접속하면 위와 같은 그누보드 초기 페이지가 나타난다.

<그림 3-19 페이지 만들기 1>



<그림 3-20 페이지 만들기 2>

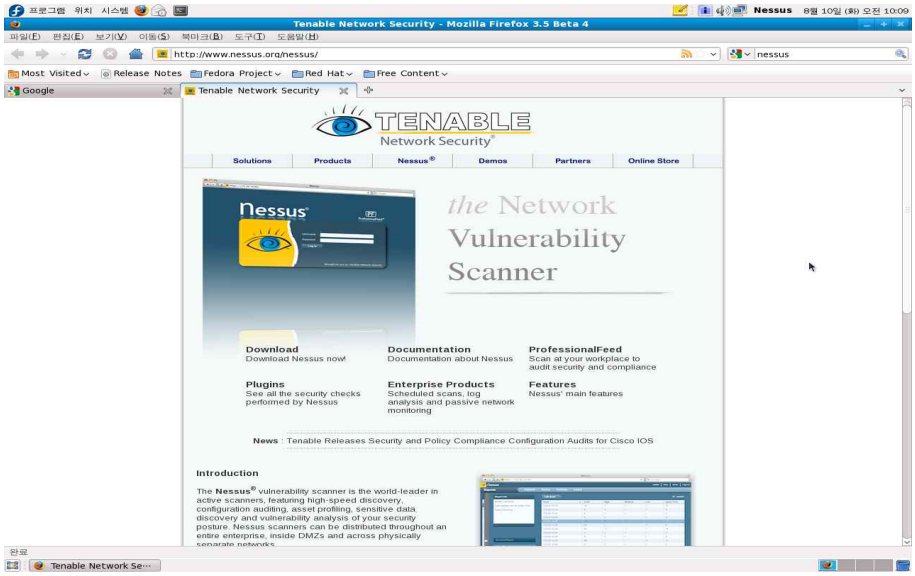
· 나모 웹 에디터를 이용하여 웹페이지를 만든다.

<그림 3-21 완성된 Web 페이지>

· 업로드를 통하여 완성된 웹페이지를 확인한 이미지는 위와 같다.

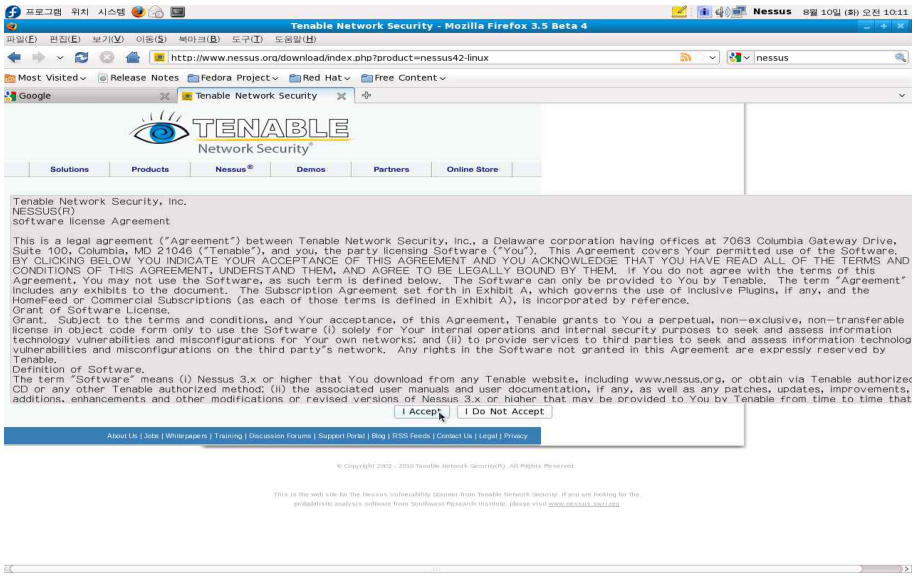
3) Nessus의 구축방법

Nessus는 Windows 버전과 Linux 버전이 있다. Linux 버전의 Nessus를 운영하려 Fedora 11을 설치하고 Nessus 홈페이지에 접속하여 RPM패키지를 다운 받아 Nessus 서버를 구축한다.
여기서 RPM이란 레드헷(Red Hat) 리눅스의 패키지를 관리해줌으로써, 사용자의 편의를 제공해주는 관리 시스템을 말한다. 리눅스는 오픈 소스 프로그램으로서 원래 소스를 컴퓨터에서 컴파일 해야 하지만, RPM명령만으로도 손쉽게 설치할 수 있게 미리 컴파일된 RPM파일들이 제공된다.



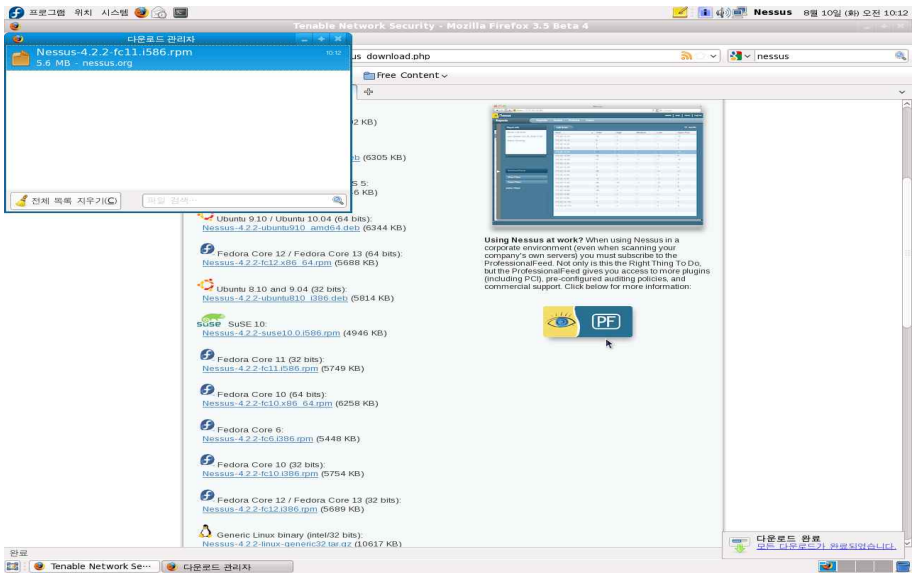
<그림 3-22 Nessus 메인 페이지>

· ‘http://www.nessus.org/nessus/’ 에 접속하면 상단에 가로로 6개의 메뉴가 있는데 그 중 Nessus를 클릭한다. 그러면 6개의 목록이 뜨는데 첫 번째 Download를 클릭한다.



<그림 3-23 Download 약관 승인>

· Download를 클릭하게 되면 약관이 뜨는데 이 약관에 동의해야만 RPM 패키지를 다운받을 수 있다. 다운 받기 위해 I Accept를 클릭한다.



<그림 3-24 Nessus Download>

· Fedora, Ubuntu, Mac OS, Windows XP, Solaris 등등 여러 종류 OS의 버전

들을 다운을 받을 수 있다. 같은 종류의 OS에서 사용할지라도 bits가 다르기 때문에 컴퓨터의 CPU를 확인하고 다운받는 것이 좋다. Fedora Core 11에 해당하는 32bits RPM패키지를 다운 받는다.

<그림 3-25 Nessus 설치>

· 터미널을 열어 'su-'라는 명령어로 root 권리자로 들어간다. 그 다음 RPM 패키지를 다운받은 폴더로 이동하여 RPM 명령어로 패키지를 설치한다.

· 명령어 : # rpm -Uvh Nessus-4.2.2.-fc11.i586.rpm

① rpm -Uvh ② Nessus- ③ 4.2.2- ④ fc11. ⑤ i586. ⑥ rpm

① 명령어 옵션 ②패키지이름 ③버전번호 ④릴리즈버전 ⑤아키텍쳐 ⑥확장자

· 명령어 Option.

-i : 패키지 설치.

-v : 설치되는 패키지 메시지 출력.

-h : 패키지를 설치할 때 해시마크(#) 출력. (진행 상황을 보여준다.)

-U : 업데이트해준다. 기존 패키지가 없으면 -i 옵션과 같다.

즉 , 패키지 설치를 하면서 업데이트를 해주고 그 과정을 출력한다는 뜻이 된다.

<그림 3-26 Nessus 계정>

· # /opt/nessus/sbin/nesuss-adduser'는 Nessus의 계정을 만들어주는 명령어다.
· ID : root , Password : qwe123 으로 했으며 또 두 번의 확인과정이 나오는데 y를 누르고 Enter를 누르면 계정이 만들어진다.

<그림 3-27 Nessus 인증 1>

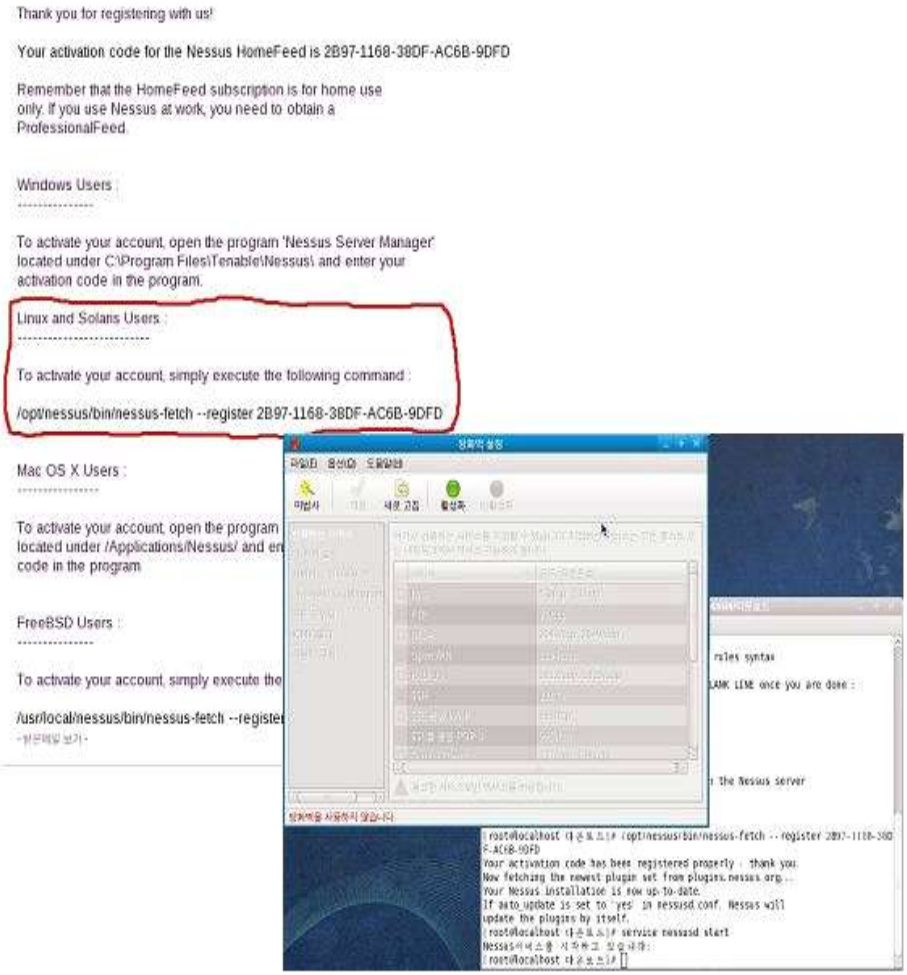
· RPM패키지를 다운 받았을 때 그 페이지의 상단에 보면 인증을 할 것이냐는 물음이 나오는데 이 인증을 받아야지만 Nessus를 작동할 수 있다.
· 문장 중 'here '이 글자를 클릭한다.

<그림 3-28 Nessus 인증 2>

· 공공기관(ProfessionalFeed)과 집(HomeFeed)에서 사용 할 수 있는 두 가지가 있는데 회사, 법인기관이 아닌 이상 HomeFeed를 선택한다.

<그림 3-29 Nessus 인증 3>

· Nessus의 인증은 메일로 받는 형식인데 인증을 받기 위해선 약관에 동의를 해야 한다. 'I Accept'를 누르면 메일 주소 쓰는 공간이 나온다. 그 곳에 자신의 메일주소를 쓰고 'Register'를 누르고 확인 화면이 뜰 것이다. 그럼 이제 적었던 메일로 접속하여 메일이 왔는지 확인을 해본다.



<그림 3-30 Nessus 인증 4>

· 메일을 확인해보면 위의 왼쪽 그림처럼 온 것을 확인 할 수 있는데 Windows XP, Linux, Mac OS, Solaris 등등 여러 OS들의 인증번호도 같이 온다. Linux를 사용하기에 Linux에 해당하는 인증번호를 복사한다.

· 복사한 인증번호를 작업하던 터미널에 복사 한 후, Enter를 누른다. 그러면 인증의 절차는 끝이 난다. 그리고 방화벽을 열어 해당 서비스를 활성화 시켜줘야 서비스를 이용이 가능한데, 다른 서비스를 이용 한다면 그에 해당하는 서비스를 체크하고 활성화 하고 나머지 불필요한 서비스는 방화벽을 막아 보안의 안전성을 높인다.

· 방화벽의 사용 선택을 끝냈으면 서비스를 시작해야한다. 그래야 설정해두었던

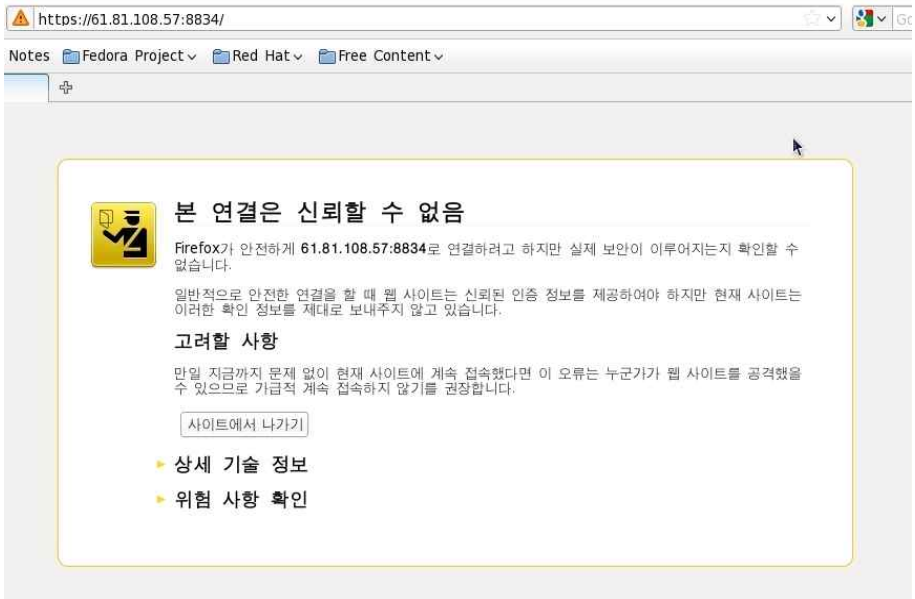
사항들이 적용되어 시작한다.

→ 명령어 : '# service nessusd start'

· 마지막에 명령어를 함으로써 이제 Fedora에서의 구축은 다 마쳤다고 보면 된다. 이제 Nessus에 접속하는 과정만 남았다. 인터넷(FireFox)으로 Nessus 서버에 접속하여 보자.

· Nessus 접속 주소 : https://IP주소:포트번호/ - 해당 컴퓨터 IP 주소와 포트번호를 이용해 접속을 하는데 기본적인 포트번호는 8834로 설정되어 있으므로 특별한 경우가 아닌 이상 포트번호는 8834로 보면 된다.

· IP주소 대신 'localhost'로 써주는 것도 괜찮다.

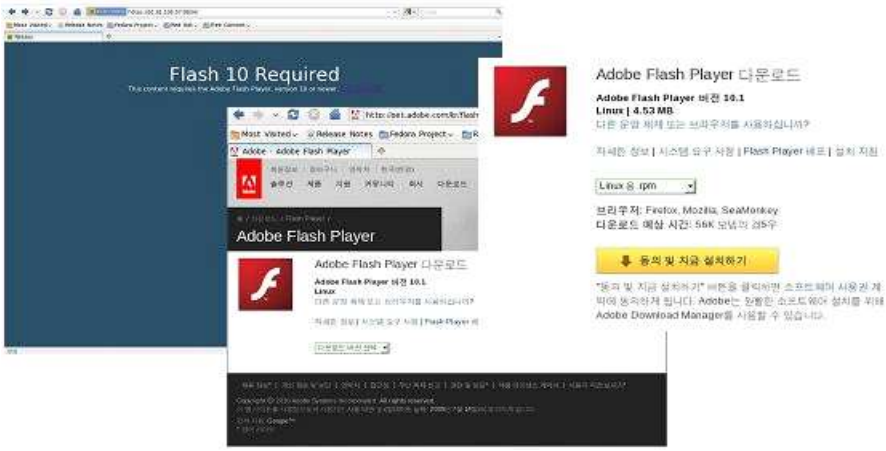


<그림 3-31 Nessus 접근 허가 인증 1>

· ' https://61.81.108.57:8834/ '로 접속하면 ' 본 연결을 신뢰할 수 없음'이라는 화면이 뜨면서 접속이 되지 않는데 이는 Nessus 서버에 접근하기 위해서는 접근 가능한 인증서가 있어야 하는데 그것이 없으므로 연결 할 수 없는 것이다.

<그림 3-32 Nessus 접근 허가 인증 2>

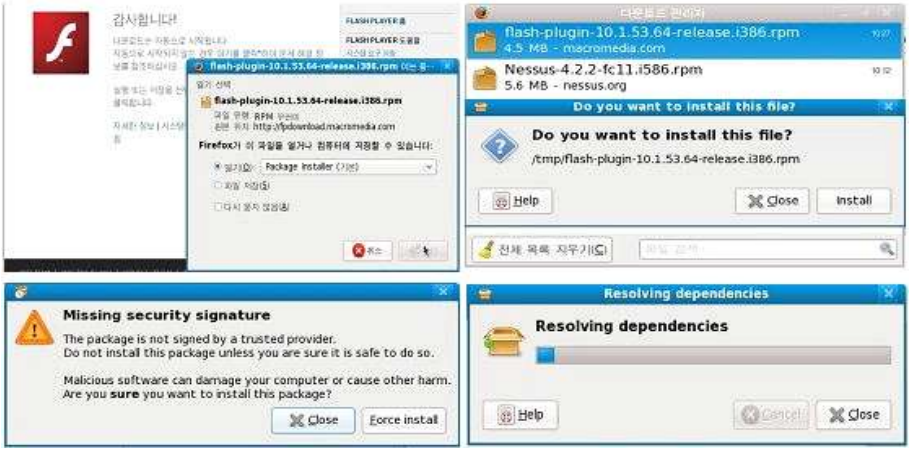
· 위험 사항 확인을 누르면 '보안 확인 예외 목록 추가' 창이 뜬다. 그 창에 '인증서 받기' 버튼을 눌러 인증서를 받고 확인을 클릭하면 인증서를 받음으로써 Nessus 서버에 접속 할 수 있게 된다.



<그림 3-33 Flash Download>

· 인증을 받았다면 다시' https://61.81.108.57:8834/ '접속해본다. 그러면 위의 왼쪽 화면처럼 나올 것이다. 이것은 Flash가 설치되어 있지 않아 화면이 보이지 않은 것인데, 보통 Flash가 설치 되어있다면 정상적인 Nessus 화면이 보여진다.

· Nessus의 접속화면이 나오도록 Flash를 다운 받아 설치 해보자. 'Get flash 10'이라는 문장을 클릭하면 'Adobe Flash Player' 다운로드 창이 뜬다. 여러 종류 중 'Linux 용 .rpm '선택. 동의 및 지금 설치를 클릭한다.



<그림 3-34 Flash 설치>

· 다운로드 창이 뜨고 저장할 폴더를 지정해준다. 다운로드 창에서 받은 RPM을 클릭하면 '설치를 원하냐'는 문구가 뜨는데 install을 클릭한다.

· 중간에 보안 신호를 잃었다는 메시지가 뜨는데 그럴 경우 강제설치(Force install)를 하면 된다.

```
[root@localhost 다운로드]# service nessusd restart
Nessus 서비스를 종료함:                                    [  OK  ]
Nessus서비스를 시작하고 있습니다:
[root@localhost 다운로드]#
```

· 명령어 : '# service nussusd restart'

→ 명령어를 이용하여 Nessus 서비스 재시작을 한다.

<그림 3-35 Nessus 접속 페이지>

· 'https://61.81.108.57:8834/'접속하면 위의 왼쪽 그림처럼 로그인 하는 화면이 뜨면 설정해 두었던 ID와 Password를 입력하고 로그인을 하면 오른쪽과 같은 Nessus 서버의 접속 화면이 나타난다. 이리하여 Nessus 의 구축은 끝이 난다.

4) Snort & Base의 구축방법

Snort와 Base는 Nessus와는 다르게 Fedora에 구축 하였으나, 알 수 없는 에러들에 의해 Ubuntu에 구축을 하게 되었다. 또 원래 Base 가 아닌 Acid 와 연동을 해야 했고, 구축까지는 했는데 마지막에 찾아도 나오지 않는 에러에 의해 Base를 구축하게 되었다.

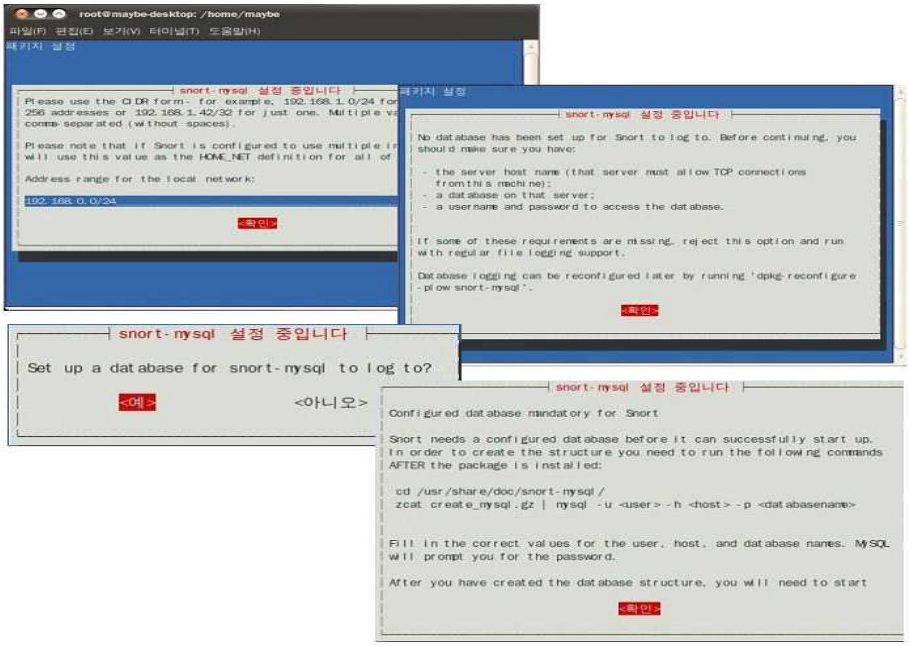
<그림 3-36 Snort 설치>

· '$ sudo su'root의 권한으로 접속한다.

· 명령어 : # sudo apt-get install snort-mysql

sudo apt-get install snort

· Fedora와는 다른 명령어를 사용하는 Ubuntu는'apt-get'란 설치 명령어로 Snort와 mysql를 설치해준다.



<그림 3-37 snort-mysql 설정 확인>

· 설치 중간에 snort-mysql 의 설정에 대한 확인절차가 나온다.

· 그 확인절차 중 IP 주소의 범위를 정할 수 있는데, 예를 들어 '192.168.0.0/16' 이런 형식으로 나온다. 개인이 사용하고 싶은 만큼의 범위를 정하는 것이라 위의 가장 왼쪽의 그림처럼 '192.168.0.0/24' 즉, 전체 네트워크를 범위로 정했다.

<그림 3-38 Snort Rule 업데이트>

· 명령어 : # sudo apt-get install oinkmaster

· snort rule의 자동 업데이트인 'Oinkmaster'를 설치한다.

· Oinkmaster는 간단하면서도 편리한 스크립트로써 snort rule을 자동 업데이트 하고 변경하기 위한 용도로써 사용된다. 또한 사용여부를 설정할 수 있으며 최신 업데이트에 대한 정보를 제공해준다.

· 명령어 : # sudo oinkmaster -o /tmp/

ls /tmp

· oinkmaster의 출력 디렉토리를 tmp라는 폴더로 설정해준다.

```
root@maybe-desktop:/home/maybe# oinkmaster -o /etc/snort/snort.conf
Loading /etc/oinkmaster.conf

/usr/sbin/oinkmaster: Error: the output directory "/etc/snort/snort.conf" doesn'
t exist or isn't readable by you.

Oink, oink. Exiting...
root@maybe-desktop:/home/maybe# oinkmaster -o /etc/snort/rules/
Loading /etc/oinkmaster.conf
Downloading file from http://www.snort.org/dl/rules/snortrules-snapshot-2_2.tar.
gz...
/usr/sbin/oinkmaster: Error: could not download from http://www.snort.org/dl/rul
es/snortrules-snapshot-2_2.tar.gz. Output from wget follows:

--2010-09-29 09:27:38--  http://www.snort.org/dl/rules/snortrules-snapshot-2_2.
tar.gz
Resolving www.snort.org... 68.177.102.20
접속 www.snort.org|68.177.102.20|:80... 접속됨.
HTTP request sent, awaiting response... 404 Not Found
2010-09-29 09:27:40 ERROR 404: Not Found.


Oink, oink. Exiting...
```

<그림 3-39 Snort Rule 경로 설정>

· 명령어 : # sudo oinkmaster -o /etc/snort/rules/

· oinkmaster를'/etc/snort/rules/'디렉토리 안에도 출력을 설정해 준다.

```
root@maybe-desktop:/home/maybe# vi /etc/snort/snort.conf
#
# var HOME_NET [10.1.1.0/24,192.168.1.0/24]
#
# MAKE SURE YOU DON'T PLACE ANY SPACES IN YOUR LIST!
#
# or you can specify the variable to be any IP
# like this:

var HOME_NET any

# Set up the external network addresses as well
var EXTERNAL_NET any
#var EXTERNAL_NET !$HOME_NET

# Configure your server lists. This allows sno
# systems that have a service up. Why look for
# running a web server? This allows quick filt
# These configurations MUST follow the same con
# above for $HOME_NET.

# List of DNS servers on your network
var DNS_SERVERS $HOME_NET
```

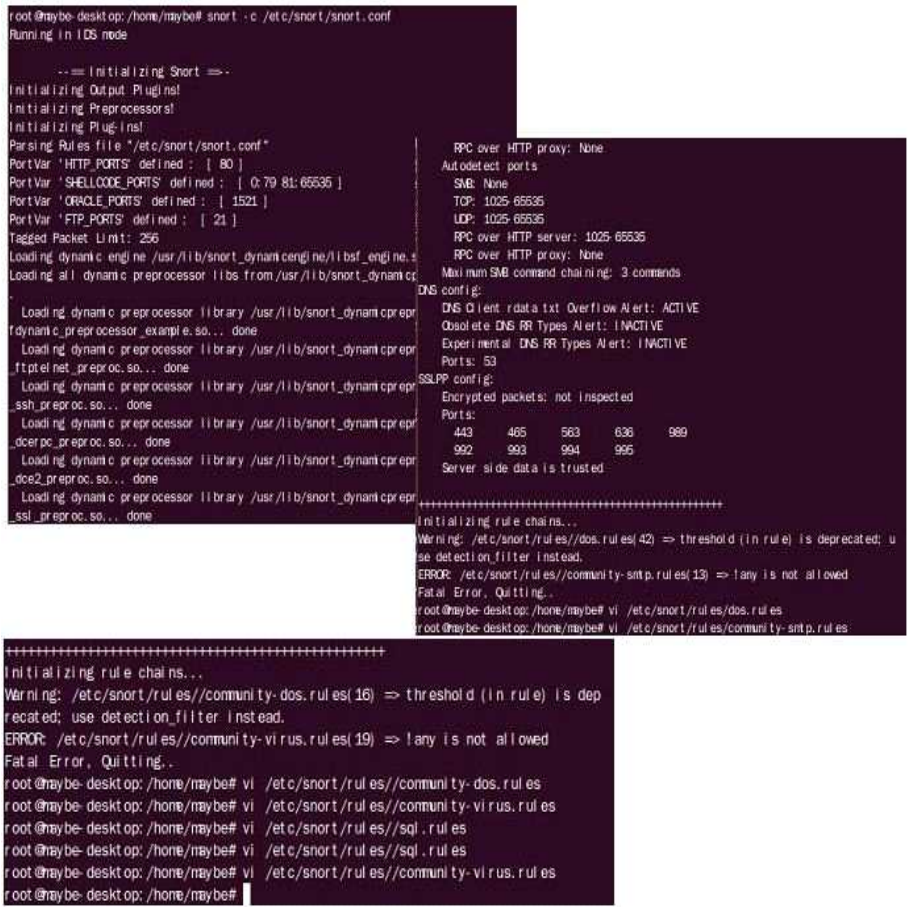
```
#
# AIM servers. AOL has a habit of adding new AIM servers, so instead of
# modifying the signatures when they do, we add them to this list of servers.
var AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12
.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.15
3.0/24,205.188.179.0/24,205.188.248.0/24]

# Path to your rules files (this can be a relative path)
# Note for Windows users: You are advised to make this an absolute path,
# such as: c:\snort\rules
var RULE_PATH /etc/snort/rules/
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# Configure the snort decoder
# ==========================
#
# Snort's decoder will alert on lots of things such as header
# truncation or options of unusual length or infrequently used tcp options
#
#
# Stop generic decode events:
#
```

<그림 3-40 snort.conf 수정>

· 명령어 : # sudo vi /etc/snort/snort.conf

· /etc/snort/ 안에 있는 snort.conf 파일의 설정을 수정한다.
· 수정 할 부분 : ‘var HOME_NET any’, ‘var RULE_PATH /etc/snort/rules/’
· ‘var HOME_NET any’⇒ snort 탐지하기 위한 범위를 설정.
· ‘var RULE_PATH /etc/snort/rules’⇒ snort rule의 변수를 설정.

※ 비슷한 부분이 많기에 찾기가 힘든 경우가 있다. 그럴 때 ‘/var RULE_PATH’ 혹은‘/var RULE_PATH /etc/snort/rules’ 적은 후 Enter 치면 찾는 부분으로 이동한다.즉 ‘/’는 찾기 기능을 한다고 보면 된다. 혹, 같은 부분이 계속 나오는데 찾는 부분이 아니라면 ‘n’을 눌러 원하는 부분을 찾는다.



<그림 3-41 snort.conf 실행 및 에러 수정>

· 명령어 : # sudo snort -c /etc/snort/snort.conf
· '/etc/snort'에 있는 snort.conf 파일로 snort 실행한다. 그러나 경고 혹은 에러

메시지가 뜨면서 특정 파일의 몇 번째 줄에 이상이 있다고 나올 것이다.

· 해당 파일을 vi 편집기를 이용하여 문장의 맨 앞에'#(주석)'을 적어 준 후, ESC 키를 누른 다음 ':wq(명령어,'저장하고 종료'뜻을 가지고 있다.)'을 적고 Enter 키를 눌러주면 파일의 편집을 마치고 나오게 된다.

· 하나의 경고, 에러를 가지고 있는 것이 아니니 에러가 뜨지 않을 때까지 '# sudo snort -c /etc/snort/snort.conf'을 실행해주면서 에러를 잡아준다.

※ 명령어 :'wq' ⇒ '저장하고 종료'뜻을 가지고 있다.

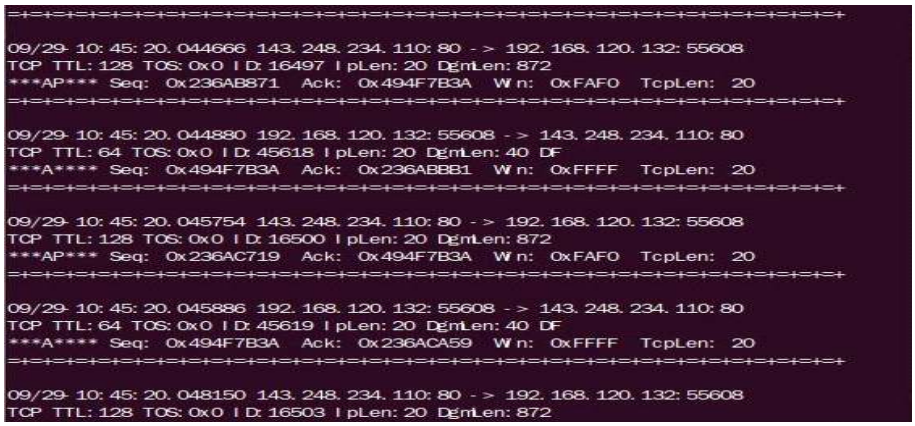
※ 몇 번째 줄인지 일일이 세기가 힘들다면 vi 편집기를 이용해 해당파일 안으로 들어가 ':set nu'를 적어주면 파일 안에 있는 문장에 순서대로 번호가 붙게 된다. 그것을 이용해 찾으면 한결 수월해진다.

Ex) Warning: /etc/snort/rules//dos.rules(42)
=> threshold (in rule) is deprecated; use detection_filter instead.

ERROR: /etc/snort/rules//community-smtp.rules(13)
=> !any is not allowed

<그림 3-42 경고 주석처리>

· 위의 경고의 주석 처리 한 것의 모습을 캡쳐한 모습.



<그림 3-43 snort.conf 실행>

· 더 이상의 에러가 나오지 않게 수정해주고 다시 한번 명령어 '# sudo snort -c /etc/snort/snort.conf'을 해주었을 때 위의 그림처럼 실시간 패킷의 캡쳐를 보여준다면 명령어를 제대로 실행한 것이다. (테스트하기 위해 경고, 에러를 없애주는 과정을 했다고 보면 된다.)

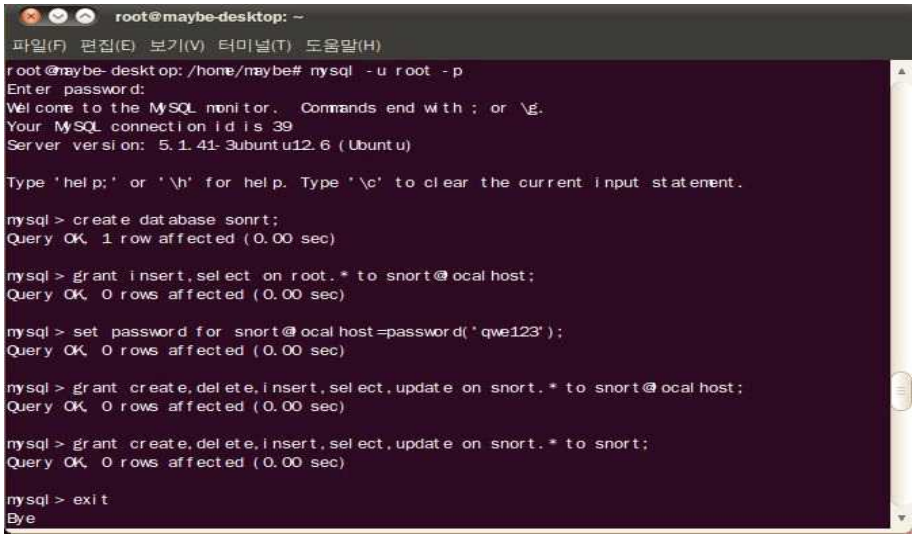
※ 테스트가 제대로 이루어진 명령어 :

/var/log/snort/alert → '/var/log/snort/alert'에 snort의 로그파일이 쌓이도록 설정해준다.

<그림 3-44 Mysql 설치>

· 명령어 : # sudo apt-get install mysql-server

· mysql-server를 설치 해주면 root의 password를 설정해주고 다시 한번의 확인절차를 해준다.



<그림 3-45 Snort 유저 등록>

· mysql 사용자 설정과 snort 유저를 등록해준다.

· # mysql -u root -p

> set password for root@localhost=password('PICK_A_PASSWORD');

→ 관리자 패스워드 설정.

(mysql-server 설치 시 password를 설정해주었다면 해주지 않아도 된다.)

> create database snort;

→ snort db 생성.

> grant insert,select on root.* to snort@localhost;

→ snort 라는 계정을 추가.

> set password for snort@localhost=password('PASSWORD_snort_CONF');

→ snort 비밀번호 설정.

> grant create,delete,insert,select,update on snort.* to snort@localhost;

→ snort 권한 설정.

> grant create,delete,insert,select,update on snort.* to snort;

→ snort 권한 설정.

> exit

· 명령어 :

mysql -u root -p

> set password for root@localhost=password('snort');
> create database snort;
> grant insert,select on root.* to snort@localhost;
> set password for snort@localhost=password('qwe123');
> grant create,delete,insert,select,update on snort.* to snort@localhost;
> grant create,delete,insert,select,update on snort.* to snort;
> exit

```
mysql > show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| snort              |
+--------------------+
3 rows in set (0.00 sec)

mysql >
```

<그림 3-46 Databases 생성 확인>

· > show databases;
· 데이터 베이스가 생성된 모습을 볼 수 있다.

```
#output log_tcpdump: tcpdump.log

# database: log to a variety of databases
# ---------------------------------------
# See the README.database file for more information about configuring
# and using this plugin.
#
 output database: log, mysql, user=root password=qwe123 dbname=snort host=localhost
# output database: alert, postgresql, user=snort dbname=snort
# output database: log, odbc, user=snort dbname=snort
# output database: log, mssql, dbname=snort user=snort password=test
# output database: log, oracle, dbname=snort user=snort password=test
# <debian>
# Keep your paws off of these (#DBSTART#) and (#DBEND#) tokens
# or you *will* break the configure process (snort-pgsql/snort-mysql only)
# Anything you put between them will be removed on (re)configure.
#
# (#DBSTART#)
# (#DBEND#)
```

<그림 3-47 Mysql과 Snort 연동>

· mysql과 연동을 위해 /etc/snort/snort.conf 파일을 vi 편집기를 이용하여 설정할 부분을 수정한다.

· Datebase 설정해준다.
· 소스 수정 할 부분.
· output log_tcpdump: tcpdump.log → # output log_tcpdump: tcpdump.log
· output database: log, mysql, user=root password=test dbname=db host=localhost
→ output database: log, mysql, user=root password=qwe123 dbname=snort host=localhost

```
root@maybe-desktop:/# mysql -u root -p < /home/maybe/snort-2.8.6.1/schemas/create_mysql snort
Enter password:
```

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql > show tables;
+-----------------+
| Tables_in_snort |
+-----------------+
| data            |
| detail          |
| encoding        |
| event           |
| icmphdr         |
| iphdr           |
| opt             |
| reference       |
| reference_system|
| schema          |
| sensor          |
| sig_class       |
| sig_reference   |
| signature       |
| tcphdr          |
| udphdr          |
+-----------------+
16 rows in set (0.00 sec)

mysql >
```

<그림 3-48 Snort 테이블 확인>

· # mysql -u root -p < /home/maybe/snort-2.8.6.1/schemas/create_mysql snort
· 'create_mysql snort'의 위치 확인 후 mysql에 snort 테이블을 생성해준다.
· # mysql -u root -p
> show databases;
> use snort
> show tables;
· mysql -u root -p 실행하여 mysql에 snort 테이블이 제대로 생성되었는지 확인해본다.

<그림 3-49 Apache, php, Adodb Library 설치>

· 명령어 : # sudo apt-get install apache2 php5-mysql libphp-adodb
· apache, php, adodb libphp 를 자동 설치해준다.

<그림 3-50 Adodb와 BASE Download>

· adodb 는 mysql 에 있는 데이터베이스 추상화 라이브러리로써 웹 브라우저에서 DB값들을 띄어주기 위해서 사용 된다 .
· 그래프를 보여주기 위해 BASE를 압축버전으로 다운로드 한다.

<그림 3-51 BASE 압축 해제>

· # sudo mv base-1.2.2.tar.gz
· # cd /var/www
· # sudo tar -xvzf base-1.2.2.tar.gz
· # sudo rm base-1.2.2.tar.gz
· # mv base-1.2.2 base
· 압축을 풀면 base-1.2.2 라는 폴더가 생기는데 인터넷에 접속할 때 간편하게

접속하기 위해 mv(=move) 라는 명령어를 이용하여 /var/www/base 이동한다.

```
$DBlib_path = '/var/www/adodb5';

$BASE_urlpath = 'base/';
```

<그림 3-52 base_conf.php 수정>

- 명령어 : # sudo vi base/base_conf.php
- vi 편집기를 이용하여 base_conf.php 파일의 부분을 수정한다.
- adodb 라이브러리 경로를 '/var/www/adobe5'지정한다.
- BASE 웹 브라우저의 경로를 'base/'로 지정한다.

```
$DBtype = 'mysql';

/* Alert DB connection parameters
 *   - $alert_dbname   : MySQL database name of Snort alert DB
 *   - $alert_host     : host on which the DB is stored
 *   - $alert_port     : port on which to access the DB
 *   - $alert_user     : login to the database with this user
 *   - $alert_password : password of the DB user
 *
 *  This information can be gleaned from the Snort database
 *  output plugin configuration.
 */
$alert_dbname   = 'snort';
$alert_host     = 'localhost';
$alert_port     = '';
$alert_user     = 'snort';
$alert_password = 'qwe123';

/* Archive DB connection parameters */
$archive_exists   = 0; # Set this to 1 if you have an archive DB
$archive_dbname   = 'snort_archive';
$archive_host     = 'localhost';
$archive_port     = '';
```

<그림 3-53 BASE 설정>

- 총 BASE의 설정 중의 한 모습이다.
- $alert_dbname = 'snort'; → db의 이름이 snort 설정.
  $alert_host = 'localhost'; → host를 localhost로 설정.
  $alert_port = ''; → port는 따로 설정해주지 않아도 된다.
  $alert_user = 'snort'; → 사용자 이름을 snort로 설정.
  $alert_password = 'qwe123'; → password를 qwe123으로 설정.

```
root@maybe-desktop:/home/maybe# snort -c /etc/snort/snort.conf
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 80 ]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1521 ]
PortVar 'FTP_PORTS' defined : [ 21 ]
Tagged Packet Limit: 256
Loading dynamic engine /usr/lib/snort_dynamicengine/libsf_engine.so... done
Loading all dynamic preprocessor libs from /usr/lib/snort_dynamicpreprocessor/..
.
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//lib_s
fdynamic_preprocessor_example.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf
_ftptelnet_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf
_ssh_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf
_dcerpc_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf
_dce2_preproc.so... done
  Loading dynamic preprocessor library /usr/lib/snort_dynamicpreprocessor//libsf
_ssl_preproc.so... done
```

<그림 3-54 Snort Background 작업>

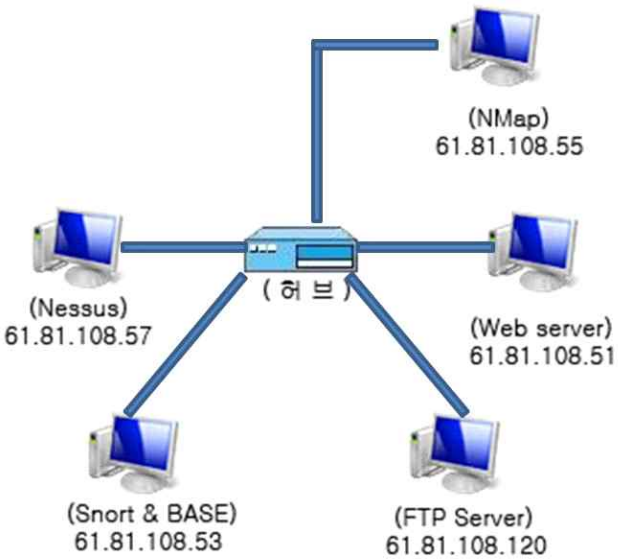
· Snort 는 현재 백그라운드로 작업 중 에 있으며 백그라운드에서 실시간 적으로 외부와 내부 네트워크를 탐지 하고 있으며 명령어 snort -c /etc/snort.conf 을 실행함으로써 /var/log/snort 에 snort 탐지에 대한 로그가 실시간적으로 생성되고 있다.

· 위의 화면이 출력됨으로써 Snort&BASE의 구축을 완료했다.

결 론

4. 결과

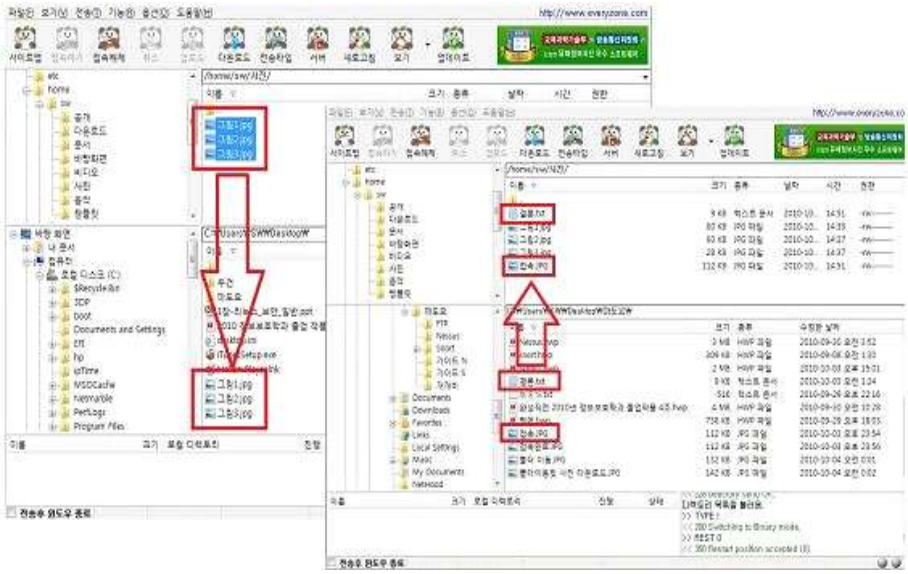
1) 시스템 구성도



· 사내망에서 FTP, Web 서버를 구성하고 Snort&BASE를 침입 탐지 시스템(IDS)으로 구성하였으며 취약점 점검을 위하여 Nessus 서버를 구성하고 있다.
· IP : '61.81.108.55' 인 PC가 Nmap을 이용하여 악의적인 공격한다 가정하여 Snort&BASE가 공격이 탐지되면 경고를 알림 후 패킷을 분석한다.

2) 정보서비스 운영 - FTP, Web

· FTP Server
FTP 서비스는 GUI(프로그램)서비스와 CUI(콘솔)서비스 두 가지를 제공한다.



<그림 4-2 FTP Upload&Downloan 확인>

· 알FTP나 에브리존FTP 등을 이용하여 해당 IP에 접속 할 수 있다. 관리자가 제한하지 않는 폴더가 아닌 이상 여러 폴더로 이동이 가능하며, 다운로드나 업로드를 막아 놓지 않는 이상 여러 파일들을 업로드 혹은 다운로드를 할 수 있으며 보통의 이러한 형태로 FTP서버를 이용 할 수 있다.

```
[root@localhost ~]# ftp 61.81.108.120 21
Connected to 61.81.108.120 (61.81.108.120).
220 (vsFTPd 2.1.2)
Name (61.81.108.120:sw): root
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

<그림 4-3 FTP CUI 접속>

· CUI(콘솔)인 ssh에서도 FTP서버 접속이 가능하다.

· Web Server

<그림 4-4 Web 페이지>

· 현재 기업은 위의 사진처럼 운영되고 있습니다.

3) Nessus 취약점 분석

· Nessus는 관리자 서버의 네트워크 취약점을 분석할 수 있다.

· 일단 관리자는 ID 와 Password로 접속한다. 컴퓨터의 취약점을 탐지하기 위해 상단의 Policies(정책설정)을 클릭하여 이름을 설정해주고 Next를 클릭하며, 다른 옵션들을 선택해주고 생성해준다.

· Scans(스캔설정)으로 이동하여 Add Scan을 클릭. 스캔하고자 하는 대상을 설정해준다. Name, Policy, Scan Targets를 설정해주는데 Name는 원하는 이름으로 해주고 Policy는 위에서 만들어주었거나 다른 것이 있다면 그것으로 설정해준다. Scan Targets은 localhost로 정해주어도 되고 다른 IP주소를 정해줘도 된다.

· 설정해주었던 IP에서의 스캔이 시작되고 있음을 보여주며 스캔이 완료되면 모습은 자동적으로 사라진다.

· Report로 이동하면 스캔이 완료되었음을 보여준다. 해당 이름 더블클릭하게 되면 간략하게 패킷에 대해 보여준다. 더 자세히 알고 싶다면 왼쪽에 Download Report을 클릭한다. 두가지의 종류가 있는데 그 중 HTML형식으로 볼수 있는 HTML.export를 선택한다.

<그림 4-5 HTML형식의 Report>

· 이 화면은 Nessus의 취약점을 HTML 형식으로 보여준 모습이다. 스캔한 시간, 취약점이 있는 포트번호, 원격 호스트 정보 등을 보여주며 그 아래로는 각 포트에 어떻게 몇 번 포트이 강하고 약한지에 대한 자세한 설명이 나온다.

· 현재 이 네트워크를 스캔한 시간은 약 1분도 채 걸리지 않았으며, 열린 포트는 17개이며, 그 중 위험 요소를 가지고 있는 포트는 44개, 중간은 45개, 가장 낮은 즉, 안정적인 포트는 70개를 가지고 있다.

Port smtp (25/tcp) [-/+]

Service Detection

An SMTP server is running on this port.

Plugin ID:
22964

SMTP Server Detection

Synopsis:
An SMTP server is listening on the remote port.

Description:
The remote host is running a mail (SMTP) server on this port.

Since SMTP servers are the targets of spammers, it is recommended you disable it if you do not use it.

Risk factor:
None

Solution:
Disable this service if you do not use it, or filter incoming traffic to this port.

Plugin output:

Remote SMTP server banner :

220 localhost.localdomain ESMTP Sendmail 8.14.3/8.14.3; Mon, 4 Oct 2010 04:11:39 +0900

Plugin ID:
10263

Remote listeners enumeration

The Linux process '/usr/sbin/sendmail.sendmail' is listening on this port.

Plugin ID:
25221

<그림 4-6 25번 포트 점검 결과>

· 예로 25번 포트의 설명이 나와 있다. 이 포트는 비교적 안정적인 포트에 속하며, 이에 대한 자세한 설명은 다음과 같다.

· Port smtp (25/tcp) [-/+]

Service Detection

AN SMTP server is running on this port

Plugin ID:22964

SMTP Server Detection

Synopsis:
An SMTP server is listening on the remote port.

Description:
The remote host is running a mail (SMTP) server on this port.

Since SMTP servers are the targets of spammers, it is recommended you disable it if you do not use it.

Risk factor:
None

Solution:
Disable this service if you do not use it, or filter incoming traffic to this port.

Plugin output:

Remote SMTP server banner :

220 localhost.localdomain ESMTP Sendmail 8.14.3/8.14.3; Mon, 4 Oct 2010 04:11:39 +0900

Plugin ID:
10263

Remote listeners enumeration

The Linux process '/usr/sbin/sendmail.sendmail' is listening on this port.

Plugin ID:
25221

· 25번 포트에 대한 설명을 해석한 것이다.

· Port smtp (25/tcp) [-/+]

서비스 탐지
이 포트는 SMTP 서버에서 실행되고 있다.

플러그인 ID : 22964

SMTP Server 탐지

개요
SMTP 서버는 원격 포트에서 수신 대기.

설명:
원격 호스트가 이 포트에서 메일 (SMTP) 서버를 실행 중이다.

SMTP 서버는 spammers의 대상이기 때문에, 만약 당신이 그것을 사용하지 않는다면 당신은 그것을 해제하도록 권고해야 한다.

위험 요소:
없음.

솔루션 :
그것을 사용하지 않거나 또는 이 포트에 필터 들어오는 트래픽이 있다면 당신은 이 서비스를 해제해야 한다.

플러그인 출력 :

원격 SMTP 서버 배너 :

220 localhost.localdomain 확장 SMTP 센드메일 8.14.3/8.14.3; 2010년 10월 4일 (월) 04시 11분 39초 +0900

플러그인 ID :
10263

원격 수신자 목록

리눅스 프로세스 '/usr/sbin /sendmail.sendmail' 가 이 포트에서 수신 대기한다.

플러그인 ID :
25221

4) Nmap 스캐닝 & Snort 실시간 탐지

리눅스 네트워크 스캐닝 프로그램인 Nmap을 이용하여 공격PC에서 해당 IP에 스캐닝 한 후, Snort와 BASE를 이용하여 패킷 분석을 해보았다.

```
root@hacker-desktop:/home/hacker # nmap -p 25,53 -sX -P0 -D 1.2.3.4,5.6.7.8 61.81.10
8.120

Starting Nmap 4.20 ( http://insecure.org ) at 2010-10-03 23:50 KST
Interesting ports on 61.81.108.120:
PORT   STATE  SERVICE
25/tcp closed smtp
53/tcp closed domain
MAC Address: 00:13:77:C7:E1:C6 (Samsung Electronics CO.)

Nmap finished: 1 IP address (1 host up) scanned in 1.301 seconds
root@hacker-desktop:/home/hacker #
```

<그림 4-7 25번, 53번 포트 스캐닝>

· 1.2.3.4와 5.6.7.8를 가짜 주소로 사용해 61.81.108.120의 25,53번 포트를 리눅스 네트워크 스캐닝 프로그램인 nmap으로 스캐닝을 해봤다. 특히 스캔 유형은 Xmas 스캔명령을 사용했는데 이유는 어떠한 비상태보존형 방화벽과 패킷 필터링 라우터도 뚫을 수 있기 때문이다.
· 견고한 방화벽의 경우에는 SYN 비트세트와 ACK가 없는 TCP 패킷은 모두 차단함으로써 들어오는 TCP 연결을 막는다. 이때 Xmas 스캔은 SYN비트를 없애고, 이런 규칙에서 자유롭기 때문에 스캐닝에 문제가 없다.
· 방화벽이 있는 포트는 스캐닝이 어려워 Xmas 스캐닝 옵션을 줘서 포트 2개를 스캔해봤는데 둘 다 사용안하고 있다. 25번 smtp 서비스 포트는 closed 상태이고, 마찬가지로 53번 DNS TCP 포트 또한 막혀있다.
· MAC Address 는 00:13:77:C7:E1:c6 (Samsung Electronics Co.) 로 스캐닝에 걸린 시간은 1.301 seconds 이다.

※ Nmap이란?

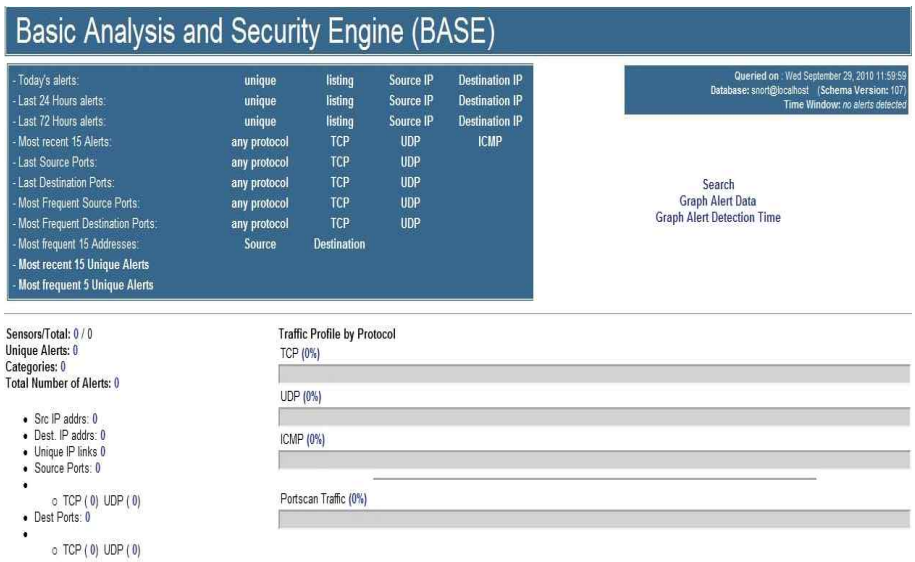
· Nmap은 네트워크 조사와 보안 감사를 위한 무료 오픈소스 유틸리티이다. 또한 수많은 시스템과 네트워크 관리자들은 Nmap이 네트워크 인벤토리와 서비스 업그레이드 스케줄 관리, 호스트나 서비스 가동 시간 모니터링 같은 작업에도 유용하다는 것을 알아냈다.

· Nmap은 어떤 호스트가 네트워크에서 사용 가능하며, 그런 호스트가 어떤 서비스를 제공하는지, 그런 호스타가 어떤 운영체제를 운영하는지, 어떤 종류의 패킷 필터/방화벽이 사용되는지 등 수십 개의 특성을 결정하기 위한 새로운 방법으로 로우 IP패킷을 사용한다.

· Nmap은 거대한 네트워크를 재빨리 스캔하려고 설계됐지만 싱글 호스트 대상으로도 잘 동작한다. Nmap은 모든 주요 컴퓨터 운영체제상에서 실행되며, 콘솔과 그래픽 버전 둘 다 사용 가능하다.

· Snort 실시간 탐지

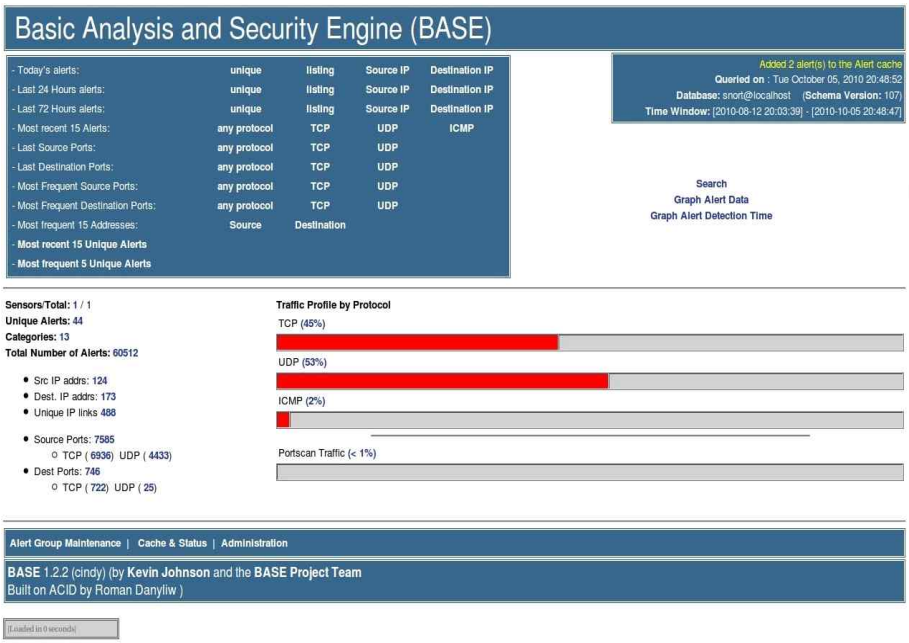
Nessus에서 네트워크의 취약점을 확인했다면 이제 Snort를 이용한 BASE의 패킷 분석을 해봤다.



<그림 4-8 Snort를 이용한 BASE의 첫 화면>

· BASE를 사용하여 분석하는 방법은 비교적 간단하다. 대부분의 인터페이스는 직관적인 화면을 제공한다.

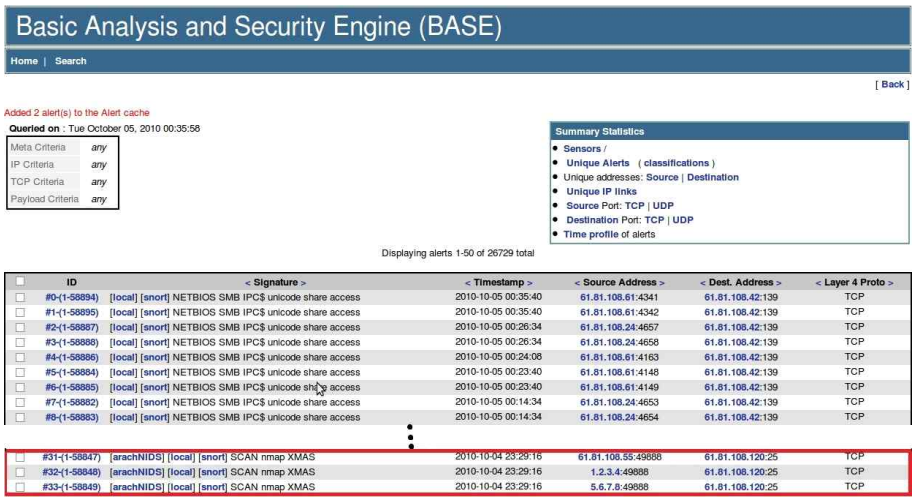
· 위의 화면은 BASE의 첫 모습이다.



<그림 4-9 BASE 일반 통계>

· 이 화면은 BASE의 일반 통계를 보여준다. 프로토콜별 경고의 수, 출발지와 목적지 포트의 수 등을 보여준다. 각 링크를 클릭하면 특정한 분류에 대한 추가적인 정보를 얻을 수 있다.

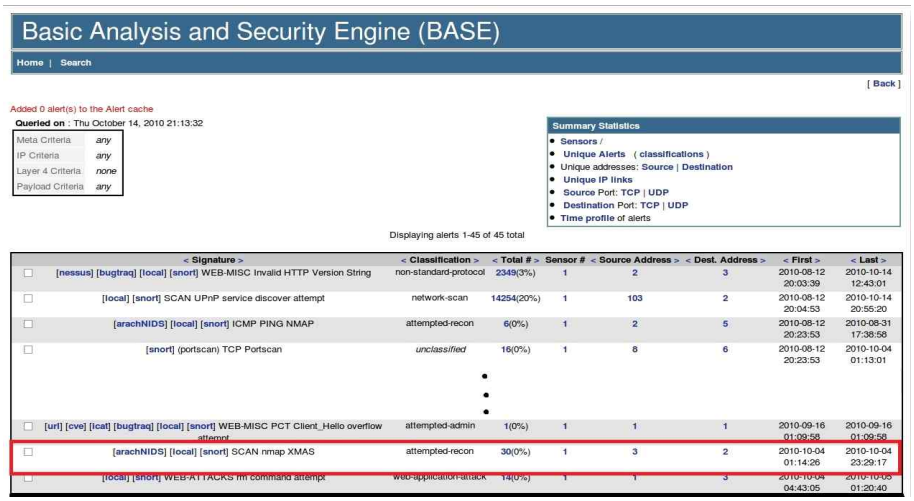
· 왼쪽 위를 보게 되면 오늘의 경보, 24시간 전의 경보, 마지막 소스 포트 등을 보여주고 있고 중간을 보면 TCP, UDP, ICMP, Portscan Traffic 종류 별로 그래프형식의 모습을 보여주는데 뭔가 접근하게 되면 그래프와 % 형식으로 탐지가 되고 있다. 위에서 스캐닝 프로그램인 nmap으로 스캐닝이 되었기에 그래프와 %가 증가한 것을 확인 할 수 있다. 그래프 왼쪽을 보게 되면 경고알림, 송신지와 수신지의 총 포트 수, 그 포트들의 TCP, UDP의 수, 송신지와 수신지의 주소 등을 확인 할 수 있다.



<그림 4-10 Nmap 스캐닝 후 Snort 탐지 모습>

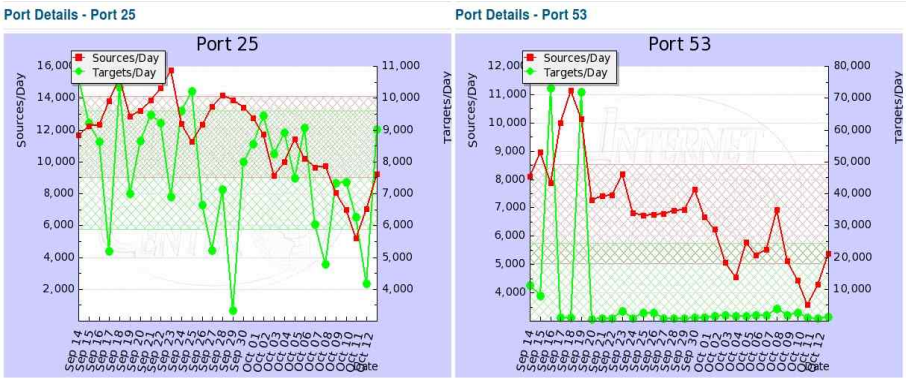
· TCP를 클릭하면 Snort에 탐지된 것을 목록으로 보여주는데 위에서 Nmap으로 주었던 포트 스캐닝이 탐지 된 것을 확인 할 수 있다.

· 여러 필드를 클릭 할 수 있는데 61.81.108.55 에서 61.81.108.120 로 언제 접근했는지 알 수 있으며 처음에 발생한 시간과 마지막으로 발생한 시간도 확인 할 수 있다.



<그림 4-11 유일한 경고 목록 >

· 그림 4-10에서 오른쪽 상단에 목록 중 'Unique Alerts'을 클릭하게 되면 유일한 경고(규칙으로 그룹화된 경고들) 목록을 보여주는데 포트 스캐닝 했던 모습이 경고 목록에 'SCAN nmap XMAS' 으로 탐지 된 것을 볼 수 있다.



<그림 4-12 BASE 그래픽화>

· BASE의 홈에 왼쪽 아래 부분을 보면 Dest. Port의 TCP를 클릭하면 여러 포트들이 나온다. 원하는 포트를 클릭하면 위의 그림처럼 그래프가 나오는데 아까 Nmap으로 스캐닝 한 25번, 53번 포트의 그래프를 볼 수 있다.

· x축은 송신지와 일을 나타내주며 적색선은 해당 포트에 하루에 얼마나 송신을 했는지 보여주고, y축은 Targets과 일을 나타내며 초록색선은 해당 포트에 Targets이 하루에 얼마나 받았는지를 보여준다.

· 25번의 포트 경우 9월 23일에 가장 많은 양을 송신하고, 10월 11일에 가장 적었다는 것을 알 수 있다. Targets은 9월 20일에 가장 적게 받았으며, 9월 18일에 가장 많은 Targets을 받았다.

53번 포트 경우 9월 18일에 가장 많은 양을 송신한 반면에 Targets은 가장 적었으며 9월 15일, 19일에 많은 것을 알 수 있다.

CVE Links

CVE #	Description
CVE-1999-9	"Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases."
CVE-1999-532	"A DNS server allows zone transfers."
CVE-1999-532	"A DNS server allows zone transfers."
CVE-1999-833	"Buffer overflow in BIND 8.2 via NXT records."
CVE-2001-10	"Buffer overflow in transaction signature (TSIG) handling code in BIND 8 allows remote attackers to gain root privileges."
CVE-2001-10	"Buffer overflow in transaction signature (TSIG) handling code in BIND 8 allows remote attackers to gain root privileges."

<그림 4-13 CVE Links>

· 이들 중에서 가장 흥미로운 것은 분류(classification) 필드와 공격 데이터베이스로의 링크(ex: Arachnids or CVE)이다. 이 데이터는 Snort가 경고를 데이터베이스에 저장할 때 규칙으로부터 가져온 것이다. 그래프 밑에 있는 CVE 링크를 클릭하면 CVE(취약점 데이트베이스)에 있는 이 공격에 대한 설명 페이지가 나올 것이다.

5) 연구 결론

요즘 전 세계적으로 빈번하게 일어나고 있는 해킹과 서비스 거부 공격이 활발해지는 가운데 우리는 자신의 정보들을 보호하지 못한채 정보들을 내주며 혹은 정보들이 빠져나간 것인지도 알지 못한 상태로 지내고 있다. 이는 그만큼 보안을 했다고 생각해도 해커들에게는 간단한 일이며, 이제는 개인의 PC를 넘어서 주로 대기업의 중요한 정보를 해킹하기 시작했다. 그 대상은 주로 관리자나 고위급의 임원이 된다.

최근 구글을 비롯 30여 기업을 공격했던 것과 같은 타겟화된 사이버공격이 새로운 방법으로 기업의 보안 모델을 시험하고 전면적인 사이버전쟁 보다 민감한 데이터에 즉각적인 위협이 되고 있다고 한다. 오래된 이메일 및 네트워크 웜, 바이러스와는 다르게 타겟공격은 도둑질 같은 것이고, 기업 네트워크에 침투해 오랫동안 숨어있을 수 있도록 하여 공격의 실제 목적인 중요한 정보를 훔치는 것이다. 타겟공격은 전형적으로 정교한 소셜 엔지니어링 기술을 이용해서 알려지지 않은 보안 취약점을 이용하는데, 일반적으로 회사에서 많이 사용하는 서명 기반의 맬웨어 추적툴을 피하도록 설계되어 있기 때문에 공격을 방어하기가 어렵다. 이를 보안하기 위해 감지가 가장 좋은 방법의 핵심이다. 타깃 공격은 네트워크를 통해서 데이터를 빼내도록 설계되어 있기 때문에, 네트워크 트래픽에 계속 주의를 기울이는 것이 방어에 도움이 될 수 있다고 한다.

앞으로도 사이버 공격은 앞으로도 계속 될 것이며 이런 공격은 모든 영역에 있는 기업들이 매력적인 공격 타겟이라는 것을 보여준다. 정확히 조준을 함으로써 치명적인 짐을 기업들에게 지어주고 있으며, 발견된 순간 이미 너무 늦은 것을 뜻 할 것이다. 기업이라면 네트워크 보안 장비를 강화 하거나 지속적인 보안 패치로 조금 더 낳은 시스템을 구축해주고 일반 사용자라면 의심스러운 IP는 리스트를 만들어 관리를 하며 만약 규칙에 어긋나는 행위를 할 경우 해당 IP를 차단해주는 등의 방법이 최선의 방법이다.

이에 우리는 지금까지 Snort와 Nessus를 이용한 안전한 기업 구축 및 운영에 대해서 연구해보았다. 물론 회사 등에서 사용하는 각종 장비들과 상용소프트웨어들과는 달리 학부 차원의 네트워크 구축 및 탐지로 모든 기업의 안전한 네트워크 구축 및 운영에 대해 알게 되었다고 할 수 없지만 가까이 학과에서부터 시작하여 더 나아가 기업의 안전한 기업 구축 및 운영이 실현될 때까지 더욱더 연구 및 학습에 노력할 것이다.

※ 별첨. IDS란?

시스템이나 네트워크 트래픽(패킷)을 모니터링 하여 자원에 대한 비정상적인 행위를 확인하고 감시하는 시스템을 뜻한다. 또한 감시를 통해 패킷에 대한 로그를 남기거나 경보를 울리도록 할 수도 있고, 일부 솔루션의 경우 방화벽과 연동하여 실시간으로 유해 트래픽을 차단하도록 할수도 있다.

◎ Pattern Matching 방식

· 장점 - 정확한 탐지가 가능하며 보안지식이 높지 않아도 쉽게 이해가 가능하고 관리자가 쉽게 룰을 편집하거나 추가/수정/삭제가 가능하다.

· 단점 - 오직 룰에 정의된 패턴만 인식할 수 있으므로 숙련된 해커의 경우 IDS 회피(evasion)가 가능하다. 또한 공격 방식이 확인된 후 룰 업데이트가 되므로 새로운 공격방식에 대한 탐지에 한계가 있으며 패턴(룰)은 수시로 업데이트 되어야 한다. 따라서 대부분의 IDS는 자동 업데이트 기능을 제공하고 있다.

◎ Anomaly Detection 방식

· 장점 - 앞의 Pattern Matching 방식과 달리 공개되지 않은 공격방식도 탐지할 수 있다. 왜냐하면 별도의 수많은 룰(패턴)을 참고할 필요없이 미리 구축된 통계 값이나 RFC 표준에 다르지 않는 경우 공격이라 판단하기 때문이며 다라서 상대적으로 탐지 속도가 빠르다.

· 단점 - 탐지할 수 있는 공격에 한계가 있으며 오탐율이 높다. 또한 통계 자료를 얻기 위해 일정 시간이 필요하다. 이를테면 갑자기 급한 일이 생겨서 새벽에도 출근할 수 있는데, 이를 단지 평소와 다르다는 이유 때문에 비정상적 행위로 판단할 수 있는 것이다.

5. 참고 문헌

· PHP가 보이는 그림책
한동순 저 | 한동순 역 | 성안당

· HTML 입문 + 웹디자인 (국배판)
성윤정 저 | 대림

· 스노트 2.0 마술상자
강유 저 | 에이콘

· Redhat Fedora
우재남 저 | 한빛미디어

· Snort&BASE 설치방법 - HOWTO: Snort Mysql Base - Ubuntu Forums
http://ubuntuforums.org/showthread.php?t=145641

· 별첨 : 리눅스 침입 탐지 시스템
이길환, 박상환 저

암호 기술 분야

• LFSR을 이용한 의사난수생성기 설계 및 구현(2조)

• 안드로이드 모바일 O.T.P 어플리케이션 및 시스템 구현(3조)

2010 졸업연구 결과보고서

LFSR을 이용한 의사난수 생성기 설계 및 구현

Design of implementation of the Pseudo-Random Generator using LFSR

팀 명	C.F. (Cyber Facilitation)
지도교수	양 정 모 교수님
	김 용 만 교수님
팀 원	김 태 준 (4학년)
	김 재 덕 (4학년)
	임 우 진 (4학년)
	김 영 식 (4학년)
	김 선 진 (4학년)

2010. 10

중부대학교 정보보호학과

요 약 문

1. 연구 제목

(국문) LFSR을 이용한 의사난수 생성기 설계 및 구현

(영문) Design of implementation of the Pseudo-Random Generator using LFSR

2. 연구 목적의 필요성

현재 시대는 인터넷이 발달됨에 따라 그 사용도가 급격하게 늘고 있다. 인터넷의 사용자가 늘어나면서 자연스럽게 정보전달의 안전성이 중요하게 되었다. 정보의 안전성은 어떤 암호알고리즘을 쓰는지에 따라서 안전성이 달라질 수도 있지만, 그 전에 더 중요한 것은 얼마나 강하고 안전한 열쇠를 사용하는지에 따라 그 암호의 장단점이 나눠진다. 우리가 이번에 연구한 의사난수 생성기법의 하나인 선형 피드백 시프트 레지스터(LFSR : Linear Feedback Shift Register)는 스트림 암호 등에 적용되는 핵심 기술이다. 이러한 의사난수 생성기법에 대한 체계적인 검정체제를 구축하고 보안성이 담보되는 응용기술을 연구하는 것은 정보보호기술 기반구조를 넓혀 나가는데 중요한 요소의 하나이다.

이번 졸업 작품에서 원시다항식을 512차까지(원시다항식을 512차까지 생성이 가능하지만, 시간이 너무 오래 걸리기 때문에 이번 졸업 작품에서는 192차까지만 원시다항식을 출력하여서 실험을 하였다.) 생성 가능한 원시다항식 프로그램을 소개하고, 임의 비트(차)범위 안에서 생성된 원시다항식을 하나 선택해서 그 원시다항식을 특성다항식으로 사용하는 LFSR에 의한 의사난수 생성기법을 컴퓨터 프로그램으로 구현하고 그 기법에 의해 생성되는 난수의 적합도를 검정할 수 있는 프로그램 5개를 개발하였다.

개발 시스템은 도수검정과 중첩도수-m검정, 비중첩도수-m 검정, poker검정, 런의 수 및 런의 길이 검정, 계열검정과 자기상관검정 5가지 검정 프로그램이다.

이번 2010년도 졸업 작품은 일정한 난수를 생성하고, 생성 된 난수를 검정해서, 의사난수의 적합성을 평가하는데 활용하는 한편 다른 의사난수 생성기법 등을 추가 개발하는 기본 시스템으로 활용하며, 의사난수 생성기의 보안성을 높이는 기술연구 기본 자료로 활용할 수 있다.

우리는 이 졸업 작품을 하는 동안 많은 어려움이 있었지만, 이렇게 최종적으로 발표를 할 수 있게 잘 이끌어 주신 교수님께 감사드리며 이 보고서를 통해서 정보보안에 대해 조금 더 관심을 가지고 보안의식이 높아지기를 바란다.

목 차

(Table of contents)

<그림 목차>

<표 목차>

1. 연구 목적 (연구의 필요성)

현재 시대는 인터넷이 발달됨에 따라 그 사용도가 급격하게 늘고 있다. 인터넷의 사용자가 늘어나면서 자연스럽게 정보전달의 안전성이 중요하게 되었다. 정보의 안전성은 어떤 암호알고리즘을 쓰는지에 따라서 안전성이 달라질 수도 있지만, 그 전에 더 중요한 것은 얼마나 강하고 안전한 열쇠를 사용하는지에 따라 그 암호의 장·단점이 나눠진다. 열쇠의 종류는 공개키 알고리즘과 비공개키 알고리즘이 있으며, 암호의 종류는 스트림암호, 블록암호, 해쉬암호로 나눌 수 있다. 알고리즘은 현재 SEED, DES, RSA, 3DES, AES 등 많이 공개가 되어있다. 우리가 이번에 연구한 의사난수 생성기법의 하나인 선형 피드백 시프트 레지스터(LFSR : Linear Feedback Shift Register)는 스트림암호 등에 적용되는 핵심 기술이다. 스트림암호 알고리즘은 주로 1970년대 초반 유럽에서 연구 발전된 LFSR(Linear Feed back Shift Register)을 이용한 이진수열 발생기를 근간으로 하여 발전하였으며, 최근에는 비트대신 워드 단위의 암호화를 하는 다양한 형태의 스트림암호가 제안되고 있다. 블록 암호가 비트들을 블록단위로 암호화하는 반면 스트림암호는 비트 단위로 암호화하므로 에러전파 현상이 없고, 일반적으로 블록 암호에 비해 속도가 빠르고 구현이 쉽다. 특히 H/W 구현시 gate 복잡도가 낮고 높은 throughput을 제공한다. 또한 스트림암호 알고리즘은 다른 암호 알고리즘과는 달리 비교적 수학적 분석이 가능하여 여러 가지 중요한 수치(주기, 선형복잡도 등)에 대하여 이론적인 값을 정확히 계산할 수 있다는 장점이 있다.[20]

이러한 의사난수 생성기법에 대한 체계적인 검정체제를 구축하고 보안성이 담보되는 응용기술을 연구하는 것은 정보보호기술 기반구조를 넓혀 나가는데 중요한 요소의 하나이다. 따라서 LFSR에 의한 의사난수 생성기법을 컴퓨터 프로그램으로 구현하고 그 기법에 의해 생성되는 난수의 적합도 등을 검정할 수 있는 소프트웨어를 개발하는 한편 LFSR의 보안 취약요인을 분석하여 보안이 담보되는 LFSR 응용 활용방안을 제시하고자 한다.

2. 연구 및 개발 방향

1) 연구개요

- 스트림암호의 구조적 특성과 이에 소요되는 난수에 대한 생성기법을 연구하고 LFSR의 이론적, 기술적인 특성을 분석하여 이를 컴퓨터 프로그램으로 구현하는 방안을 중점 연구
- 아울러 생성된 난수의 적합도 등을 검정하는 소프트웨어를 개발하고, LFSR의 보안 취약요인을 분석하고 보강방안을 연구하여 보안성을 높일 수 있는 대안을 제시

2.1 암호의 분류

암호는 크게 대칭키 암호시스템(symmetic single-key cryptosystem; SKC)과 공개키(비대칭키) 암호시스템[public-key (asymmetric) ciphersystem; PKC], 두 가지로 분류할 수 있고, 대칭키 암호시스템은 다시 스트림암호시스템(stream cipher system)과 블록 암호시스템(block cipher system)으로 나눌 수 있다.[18]

대칭키 암호를 사용하여 통신을 하고자 할 때는 송신자와 수신자가 서로 공유하고 있는 키가 필요하다. 이 키는 암·복호화에 모두 사용되므로 제 3자에게 노출되어서는 안 되는 비밀키이다. 이런 의미에서 대칭키 암호를 비밀키 암호라고도 부른다. 비밀키 암호는 어떤 두 사람 사이에도 하나의 비밀키가 필요하다. 따라서 시스템에 n명의 사용자가 존재할 경우 사용자간 암호화 통신을 위해서는 n*(n-1)/2 개의 비밀키가 있어야만 한다.[2,24]

방 법	대칭키 암호화	공개키 암호화
키 관 계	암호화키 = 복호화키	암호화키 ≠ 복호화키
키 분 배	비밀	공개 또는 비밀
키 의 수	단일키($_nC_2$)	공개키와 개인키의 쌍 (2^n)
수행속도	고속	저속
전자서명	불가능	가능
키 보 호	노출과 변경으로부터 보호	개인키에 대해서는 노출과 변경, 공개키에 대해서는 변경으로부터 보호
사용방법	메시지의 암호화	해시함수를 이용한 전자서명

[표 1. 대칭키와 공개키의 암호화 방법비교]

2.1.1 블록암호

기밀성 제공 측면에서 암호화하면 가장 먼저 떠오르는 것이 바로 블록 암호이다. 그러나 블록암호는 메시지의 기밀성 이외에도 메시지 인증이나 데이터 무결성, 심지어는 전자서명까지도 사용할 수 있다.[(2)] 이 시스템은 블록의 크기가 정해져 있기 때문에 기호의 삽입 또는 제거가 불가능하고 다양한 운영방식에 의하여 1949년 Shannon이 발표한 혼돈과 확산이론을 기반으로 설계할 수 있다는 것이 장점이다. 즉, 암호문 비트들의 통계적 분포가 평문 비트들의 통계적 분포에 어떻게 의존되는지의 판단을 어렵게 하는 혼돈(confusion)이론과 평문비트들이 암호문비트들에 어떤 영향을 주는지의 판단을 어렵게 만드는 확산(diffusion)이론을 기반으로 설계되어 암호문의 비도를 높게 할 수 있다는 것이다. 그러나 블록암호는 블록단위로 암호화되기 때문에 평문비트들을 완전히 하나의 블록으로 구성한 다음에 암호화해야 하므로 암호화과정이 블록의 크기에 따라 지연될 수 있으며, 암호화과정에서 발생하는 오류는 여러 가지 함수변환에 영향을 미치므로 오류전파가 크다는 것이 단점으로 지적되고 있다. 블록암호에 있어서 출력블록의 각 비트는 입력블록과 열쇠의 모든 비트에 의존하여 결정되도록 암호화하며 입·출력 블록의 크

기는 구현상 효율성과 보안상 안전성에 크게 의존한다.[9] 블록암호 알고리즘을 특징짓는 요소로는 Block Size, Key Size, Round-Key Generation, Round Function, Number of rounds로 전체 블록암호의 안전성이 결정된다.[2] 블록암호는 64bit, 128bit를 한 블록으로 사용하며, 블록 암호는 고전 블록과 현대 블록으로 분류된다.[2,18]

가. 고전 암호블록(대체 및 치환 암호)

- 고전 블록암호에는 모든 암호 기술의 기초가 되는 두 가지 기법인 대체와 치환암호가 있다.

(1) 대체 암호

- 대체 암호방식은 치환 암호방식이라고도 하는데 평문의 문자를 다른 문자나 숫자 또는 기호로 대체시키는 방법이다. 그 방법들로는 Caesar, 빈도수 분석, ENIGMA 등이 있다.

① <Caesar암호>

역사적으로 Caesar가 최초로 대체암호를 사용했다고 기록되어 있다. Caesar암호는 단순한 암호화 방법이다. 암호화하려는 메시지의 문자 하나하나를 암호화된 알파벳 문자로 하나씩 치환한다. 그 시대의 치환형 암호에 대한 암호 해독은 단순했다. 암호화키로 사용할 수 있는 키가 25개 밖에 없었으므로 암호를 해독하려는 사람은 25개의 키들을 하나씩 시도해봐서 숨긴 메시지를 알아낼 수 있었다.

② <문자가 사용된 빈도수를 분석하는 방법>

영어에는 문자"e"가 가장 많이 쓰이기 때문에 암호화 된 메시지에서 가장 많이 쓰였던 문자는 "e"일 거라고 생각 할 수 있다. 일반적인 영어 문장에서 "e"라는 문자가 12.7%, "z"라는 문자는 0.1% 정도의 비율로 등장한다고 한다. 암호문의 알파벳들을 검사하여 12.7%와 가장 비슷한 확률로 나온 문자는 "e"일 확률이 높

은 것이다. 즉, 언어의 통계적 성질을 이용하여 문자 빈도수에 따라 암호문 문자를 평문 문자로 대칭시킴으로써 암호문 해독이 가능하게 된다.

③ <ENIGMA>

다중 문자(Polyalphabetic) 치환 방법을 사용한 ENIGMA는 1910년부터 ENIGMA의 초기 개념을 독일, 미국, 네덜란드, 스웨덴 등지에서 연구되어진 것으로 알려졌다. 그 후 1923년 독일 출신의 엔지니어 Arthur Scherbius는 폴란드 근교의 한 시골에서 ENIGMA를 만들었다. 1차 세계대전 때는 암호문의 중요성이 널리 인식되지 않아 잘 사용하지 않았으나 2차 세계대전 많이 사용하였다. 톱니바퀴들이 복잡하게 연결된 ENIGMA는 특정 문자를 같은 의미로 두 번 이상 사용하지 않도록 교묘하게 조합되어 있었고, 더구나 조합 규칙도 매일 바뀌었다.

(2) 치환 암호(Transposition Cipher)

- 치환 암호는 평문을 고정된 길이의 블록으로 나눈 후에 각 블록 내에 있는 문자들을 일정한 방식에 의해 재배열함으로써 암호문을 생성한다. 이 중에 가장 간단한 방식은 RAILFENCE 기법으로서 평문을 대각선으로 써 놓고 횡으로 읽어내는 방식이다.

나. 현대 블록암호

- 현대 블록암호 알고리즘의 대부분은 대치와 치환의 반복에 의하여 강력한 암호 알고리즘이 설계될 수 있다는 Shannon의 이론에 근거하여 설계되었다. 현대 블록암호의 알고리즘 구조는 Feistel Network과 SPN (Substitution- Permutation Network)으로 나뉜다.

(1) Feistel 구조

- 입력을 좌우 블록으로 분할하여 한 블록을 라운드 함수에 적용시킨 후의

출력 값을 다른 블록에 적용하는 과정을 좌우블록에 대해 반복적으로 시행하는 방식으로, 라운드 키가 역순으로 작용한다는 점을 제외하면 암·복호화 과정이 동일하고 라운드 함수에 대한 제약 조건이 없어 알고리즘으로 DES, LOKI, CAST Blowfish, MISTY, RC5, CAST256, E2, Twofish, RC6, Mars 등에서 사용되고 있다.

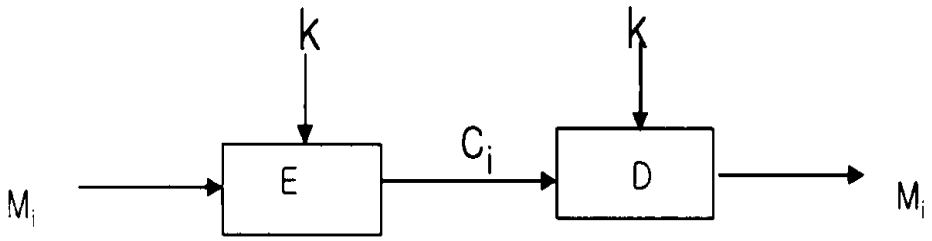
(2) SPN(Substitution- Permutation Network)

- 입력 전체 블록에 대해 한 번에 변화시키는 SPN구조는 라운드 함수가 역변환이 되어야 한다는 등의 제약이 있지만 더 많은 병렬성(parallelism)을 제공하기 때문에 암·복호화 알고리즘의 고속화가 요구되고 최근의 컴퓨터 프로세스(CPU)가 더 많은 병렬성을 지원하는 등의 현 추세에 부응하는 방식이라 할 수 있다. SPN을 사용하는 알고리즘은 SAFER, IDEA, SHARK, Square, CRYPTON, Rijndael, SAFER+, Serpent 등이 있다.

다. 블록암호 모드(운용방식)

평문의 길이가 블록암호의 길이보다 클 경우에 블록암호의 모드란 것을 사용한다. 블록암호의 주요 모드는 ECB, CBC, CFB, OFB, CTR로 나눌 수 있다.[17]

(1) ECB방식



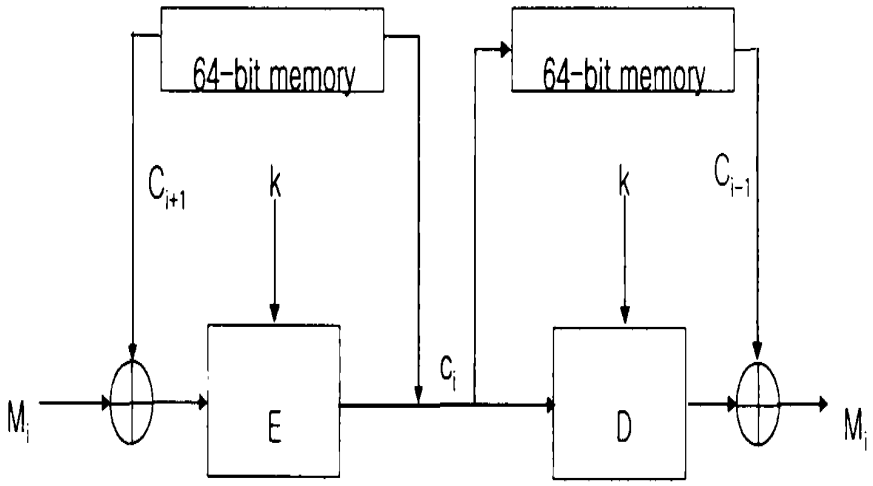
[그림 1. ECB(electronic codebook)방식에 의한 암호화와 복호화]

ECB암호화 : $C_i = E_k(M_i)$, $i = 1,2,3,...$

ECB복호화 : $M_i = D_k(C_i)$

가장 일반적인 방식으로 이 방식은 매우 단순한 방식인만큼 보안적인 측면에서 취약적인 요소를 내포하고 있다. 즉, 동일한 비밀키를 사용하는 경우 평문블록은 동일한 암호문블록으로 암호화된다는 것이다. 하지만 한 암호블록에서 일어난 오류는 다음 암호블록에 영향을 미치지 않는다는 장점이 있다.

(2) CBC방식



[그림 2. CBC방식에 의한 암호화와 복호화]

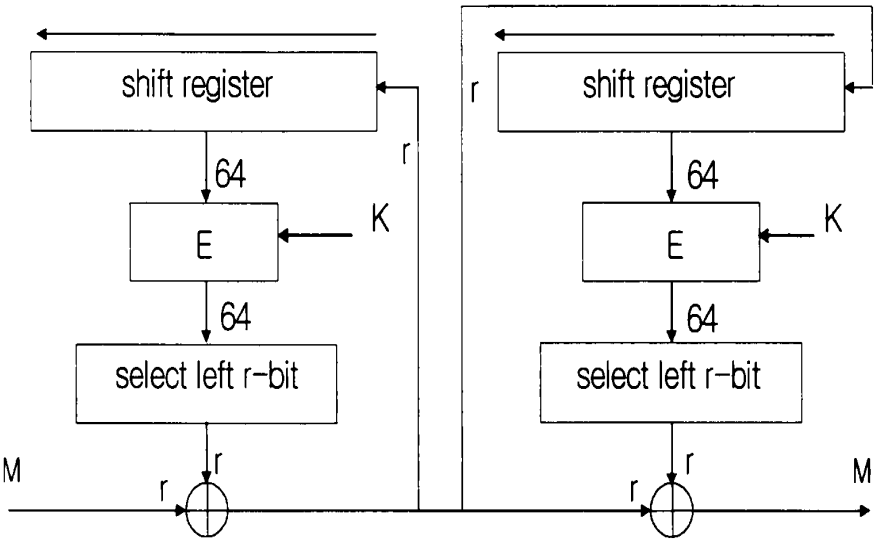
CBC암호화 : $C_i = E_k(M_i \oplus C_{i-l})$, $i = 1, 2, 3, \cdots$

CBC복호화 : $M_i = D_k(C_i) \oplus C_{i-l}$

CBC(cipher block chaining)방식은 평문 M이 64비트 블록 M_1 ,M_2 ,M_3 …….과 같이 나누어질 경우에 위와 같이 i번째 평문블록을 암호화하는 데에 i-1번째에서 생성된 암호문블록을 사용한다. 따라서 생성되는 각각의 암호문블록은 단지 현재 평문뿐만 아니라, 그 이전의 평문들의 영향도 받게 되므로 ECB 방식의 취약점

을 해결한 방식이다. 첫 번째 평문의 암호화 $E_k = (M1 \oplus C_0)$에서 C_0 는 정의되어지지 않기 때문에 맨 처음 블록에 대해서는 C_0대신에 초기화벡터(initialization vector)를 사용한다. CBC방식은 현 단계에서 생성되는 암호문블록이 그 다음으로 생성되는 암호문블록에 영향을 미치기 때문에 특정 암호문 블록이 전달되는 과정에서 발생되는 채널상의 잡음에 의한 오류가 해당 암호뿐만 아니라 그 다음 암호문에도 그 효과가 연장된다는 단점을 가지고 있다.

(3) CFB방식

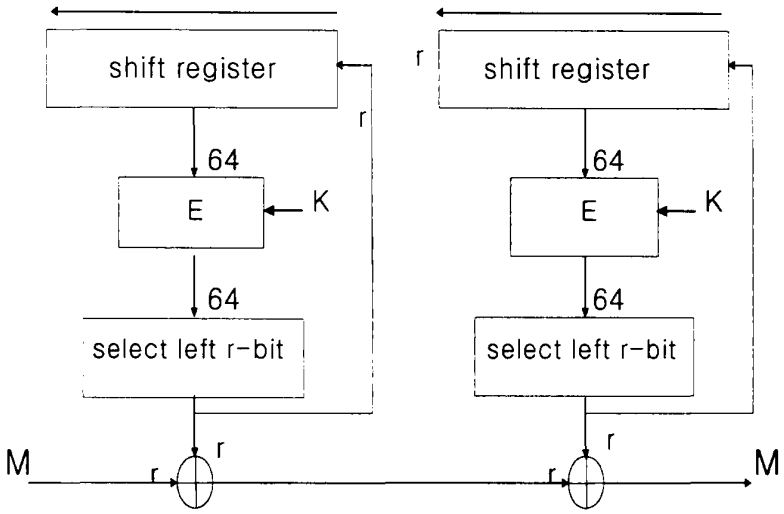


[그림 3. CFB방식에 의한 암호화와 복호화]

컴퓨터 및 통신시스템내에서의 데이터는 프레임이나 블록단위뿐만이 아니라 문자나 비트단위로도 취급되어 질 수 있는데 이 경우에는 CFB(cipher feedback)방식에 의해서 암호화가 수행될 수 있다. [그림3]은 r비트 단위로 암호화하는 CFB방식을 보여주고 있다. 맨 처음으로 CFB방식이 가동될 때에는 쉬프트레지스터안에 초기화 벡터IV를 입력하여 DES알고리즘을 수행하고 그 결과의 왼쪽r비트를

r비트평문과 XOR하여 r비트암호문을 생성하게 된다. 각각의 r비트에 대하여 암호화가 수행된 후에는 그 r비트의 암호문이 다시 키 수열을 생성하는 알고리즘에 유입된다. 유입될 때마다 쉬프트 레지스터는 r비트씩 왼쪽으로 이동한다. 결국, 이 방식은 블록암호인 DES알고리즘을 통해서 생성되니 키를 이용하는 일종의 스트림암호이다. 이 방식 또한 CBC방식과 같이 오류의 파급효과가 있다.

(4) OFB방식



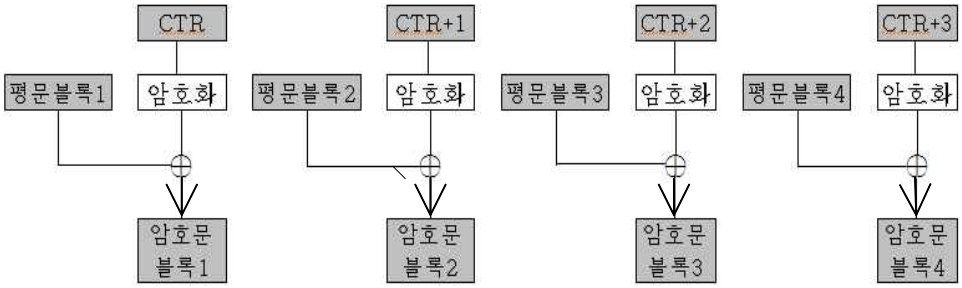
[그림 4. OFB방식에 의한 암호화와 복호화]

OFB(output feedback)방식은 CFB방식과 마찬가지로 DES알고리즘을 이용한 알고리즘에 의해서 생성되는 일종의 키 수열과 r비트 단위의 평문이 XOR되어져 암호화가 이루어진다. r비트의 문자의 암호화에 스트림암호가 적용된다는 측면에서는 CFB방식과 동일하지만 키 수열은 평문과는 무관하게 생성되어지기 때문에 현재 생성된 암호문이 그 다음 암호화에 영향을 미치지 않는다는 차이점을 가지고 있다. 맨 처음으로 OFB방식이 기동될 때에는 쉬프트 레지스터 안에 초기화벡터를 입력하여 DES알고리즘을 수행하고 그 결과의 왼쪽비트를 r비트 평문과 XOR하여 r비트의 암호문을 생성하게 된다. 각각의 r비트에 대하여 암호화가 수

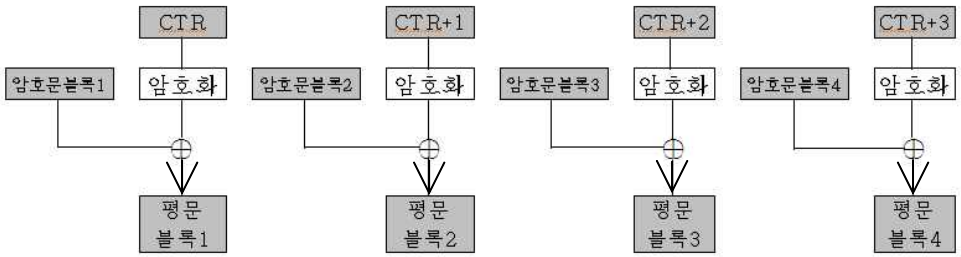
행됨과 동시에 DES알고리즘의 결과 출력되는 왼쪽의 r비트가 다시 키 수열을 생성하는 알고리즘에 유입된다. 유입될 때마다 쉬프트 레지스터는 r비트씩 왼쪽으로 이동한다.[24]

(5) CTR방식

[CTR 암호화]



[CTR 복호화]



[그림 5. CTR모드의 암・복호화]

CTR모드는 CounTeR모드의 약자이다. CTR모드는 1씩 증가해 가는 카운터를 암호화해서 키스트림을 만들어내는 스트림암호이다. CTR모드에서는 블록을 암호

화할 때마다 1씩 증가해 가는 카운터를 암호화해서 키스트림을 만든다. 즉, 카운터를 암호화한 비트열과 평문블록과의 XOR을 취한 결과가 암호문블록이 된다.

CTR모드는 OFB모드와 같은 스트림암호의 일종이다. OFB모드에서는 암호화의 출력을 입력으로 피드백하고 있지만 CTR모드에서는 카운터의 값이 암호화의 입력이 된다. 또한 이 모드에서는 암호화구조와 복호화구조가 완전히 같으므로 프로그램 구현이 매우 간단하다. 이 특징은 스트림암호의 특징과 같다. 이 모드에서는 블록을 임의의 순서로 암・복호화 할 수 있다. 암・복호화 때에 사용하는 카운터는 비표와 블록번호로부터 금방 구할 수 있기 때문이다. 블록을 임의로 처리할 수 있다는 것은 처리를 병행할 수 있다는 것을 의미하므로 병렬처리가 가능한 시스템에서는 CTR 모드를 이용하여 자료를 고속으로 처리할 수 있게 된다. 이 모드에서의 암・복호화 과정은 그림(5)와 같다.[22]

2.1.2 스트림암호(Stream Cipher)

가. 스트림암호

스트림암호는 주로 유럽을 중심으로 발전되었으며 블록 암호에 비하여 알려진 경우가 매우 적다. 그 이유 중의 하나는 대부분의 스트림암호는 H/W로 구현되며 민감한 부분에 적용되기 때문에 사용자 측면에서는 알고리즘의 형태에 대하여 알아야 될 필요가 없기 때문이다. 또한 스트림암호는 보통 긴 주기를 가지는 수열을 발생하여 문자열과 EXOR(Exclusive-OR) 하여 암호문을 발생하는 방법을 사용하기 때문에 암호의 안전성은 전적으로 수열의 안전성에 기인한다.

스트림암호 알고리즘은 주로 1970년대 초반 유럽에서 연구 발전된 LFSR(Linear Feedback Shift Register)을 이용한 이진수열 발생기를 근간으로 하여 발전하였으며, 최근에는 비트대신 워드 단위의 암호화를 하는 다양한 형태의 스트림암호가 제안되고 있다.

블록 암호가 비트들을 블록단위로 암호화하는 반면 스트림암호는 비트 단위로 암호화하므로 에러전파 현상이 없고, 일반적으로 블록 암호에 비해 속도가 빠르고 구현이 쉽다. 특히 H/W 구현시 gate 복잡도가 낮고 높은 throughput을 제공한다.

또한 스트림암호 알고리즘은 다른 암호 알고리즘과는 달리 비교적 수학적 분석이 가능하여 여러 가지 중요한 수치(주기, 선형복잡도 등)에 대하여 이론적인 값을 정확히 계산할 수 있다는 장점이 있다. 반면 블록 암호에 비해 응용분야가 제

한되어 대용량 데이터의 암호화에 주로 사용되어 왔으며 공개된 알고리즘이 제한적이여서 연구에 한계가 있어왔다. 하지만 최근에는 대수적 공격의 등장으로 많은 스트림암호가 공격당하면서 다시 스트림암호에 대한 연구가 활발해지고 있다.[20]

스트림암호는 의사난수생성기를 통해서 생성된 의사난수열과 평문을 이진수로 나타낸 값을 이진 합산해서 얻어진다. 즉, 평문 $m_1 m_2 \cdots$. 과 2진수의 비밀키 $k_1 k_2 \cdots$.를 블록 암호와는 달리 각 비트마다 XOR연산 $C_i = m_i \oplus k_i$을 하여 암호문 $C_1, C_2 \cdots$을 얻고, 복호화는 암호문의 각 비트마다 비밀키의 각 비트를 XOR 연산을 하여 평문 $m_i = c_i \oplus k_i = (m_i \oplus k_i) \oplus k_i$ 을 구하는 암호시스템을 말한다. 이때 난수 생성기를 이용해 만든 비밀키의 길이가 평문의 길이와 같고, 한번 사용한 비밀키를 다시 사용하지 않으면 완전한 암호가 된다. 이와 같이 한 번 쓴 비밀키는 버리므로 이 암호를 일회용 암호(one-time pad)라 한다.

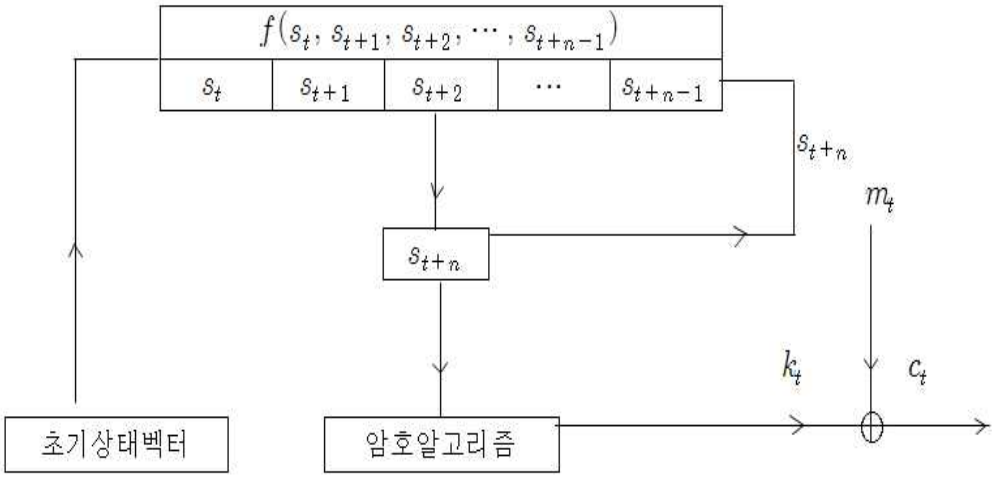
현재 스트림암호의 보편적인 흐름은 짧은 길이의 키를 공개키 암호 방식을 채택하여 키를 공유하고 암호알고리즘(대개 대칭키 암호알고리즘)을 이용하여 난수를 생성한 후 선형 귀환함수나 기타 수학적 함수들에 의해 난수열의 재배열을 이룬 후 평문과의 XOR을 통한 암호문을 생성한다. 복호화는 같은 키에 의한 같은 알고리즘의 적용으로 XOR을 시행하여 이루어진다.[21]

난수발생기는 적은 비용과 비교적 짧은 길이의 수열(초기상태)로부터 긴 길이의 이진 수열을 매우 빠른 속도로 생성할 수 있기 때문에 실제 응용에 아주 적합하다. 스트림암호 시스템을 이용하면 연설 등을 실시간으로 암호화하는 것이 가능하다. 스트림암호의 또 다른 이점은 암호문을 전송하는 동안에 생기는 에러의 발생에 덜 민감하다는 사실이다. 암호문의 한 비트가 전송도중에 변조되면 해당되는 한 비트의 복호화에만 영향을 미친다. 그러나 블록 암호로 암호화할 경우 한 비트의 오류는 오류가 생긴 비트를 포함하는 블록 전체의 복호화에 영향을 미친다. 출력 블록의 한 비트는 대응하는 입력블록의 모든 비트에 의존하기 때문이다. 그러나 이것이 블록 암호의 이점일 수도 있다. 예를 들어 공격자가 한 비트를 변경할 경우 복호화한 결과는 해당 블록 전체에 영향을 미치게 되므로 변경 사실이 쉽게 탐지될 것이다.[20]

나. 스트림암호의 운용방식

(1) 동기식 스트림암호

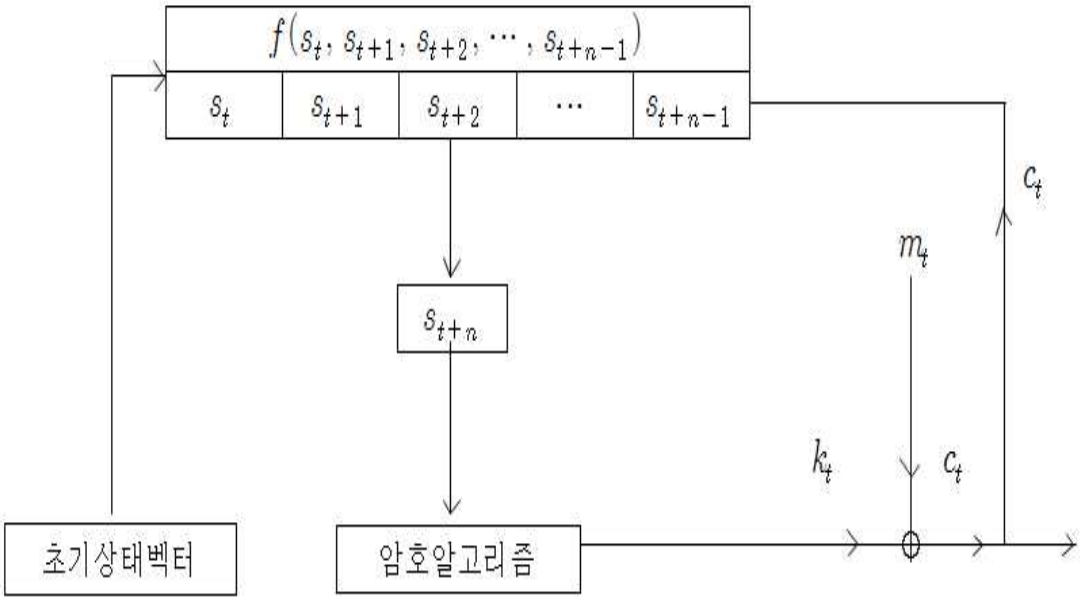
동기식 스트림암호는 주어진 초기상태 벡터로 LFSR에 의해 생성된 이진수열의 비트가 피드백 됨으로써 다음 비트를 생성하는 열쇠이진 수열로 평문을 암호화하는 스트림암호를 말한다. 여기에서 생성되는 열쇠 이진수열은 평문 이진수열과 암호문 이진수열과는 독립적으로 생성되므로 송신자와 수신자간에는 사전에 약속된 비밀열쇠와 초기상태벡터를 공유하여야만 암호화와 복호화가 가능하도록 설계된 암호시스템이다. 이 때문에 송신자가 수신자에게 암호문을 전송하는 도중에 제 3자가 1비트를 삽입하거나 삭제를 하게 되면 수신자는 잘못된 암호문을 수신하게 되어 복호를 할 수 없게 된다. 따라서 이런 문제점을 보완, 방지하기 위해서 송신자와 수신자는 일정한 간격을 두고 수시로 상태벡터를 확인하는 과정이 필요하다.



[그림 6. 동기식 스트림암호 시스템]

(2) 자기 동기식 스트림암호

자기 동기식 스트림암호는 비밀열쇠와 이진수열 S_{t+n}에 의하여 열쇠수열이 생성되는 점은 동기식 스트림암호와 같지만 수열생성자의 현재 상태를 생성할 때에는 암호문 비트가 삽입된다는 점이다. 암호문이 전송 중에 비트의 삽입과 누락이 이루어져도 n비트의 암호문이 처리된 후에는 정확하게 복호가 이루어지므로 전송 중에 오류가 전파되지 않는다.[22]



[그림 7. 자기동기식 스트림암호 시스템]

다. 안전한 스트림암호의 요구조건

스트림암호에서의 열쇠는 스트림암호시스템의 안전성을 보장하기 위하여 비예측 특성을 가져야(unpredictability)한다.

키수열이 비예측성 특성을 갖기 위해서는 최대주기, 선형복잡도, 난수성, 상관특성의 우수, 퇴화성이 없어야한다.

(1) 스트림암호에서의 열쇠는 최대주기(maximum period)를 가져야 한다. 최대주기는 $2^n - 1$ 로 구할 수 있다.

(2) 선형복잡도(linear complexity)

선형복잡도가 k이면 연속된 2k개의 수열만 알면 그 다음 수열을 예측할 수 있다. 그러므로 키 수열의 복잡도가 높다는 의미는 보다 많은 키 수열의 비트들이 관측되어야 그 키 수열을 생성하는 선형 점화식을 계산해 낼 수 있음을 의미한다.[25]

(3) 난수성(randomness)

- 1과 0의 평균횟수가 동일하여야 한다.
- “0101010101…….”과 같은 형태가 길지 않아야 한다.
- 간단한 알고리즘으로 설계와 구현이 쉬워야 한다.
- 이미 나온 출력으로부터 그 전의 값을 유추할 수 없어야 한다.
- 패턴과 역 상관관계를 알 수 없어야 한다.
- 긴 주기를 가져야한다.[21]

(4) 상관특성이 우수해야한다. 결합함수 뿐만 아니라 알고리즘 구조적인 면에서도 발생하며 많은 스트림암호가 상관특성으로부터 취약점을 갖는다.

(5) 퇴화성(Degeneracy)이 없어야 한다.

2.2 의사난수 생성기법

보안 시스템은 암호화 기능을 필요로 하고, 암호화를 위한 비밀키로 난수를 사용한다. 전자 서명에서의 비밀 파라미터나 각종 인증 메커니즘에서 세션 키, 암호화 알고리즘과 네트워크 프로토콜의 초기 벡터에 난수가 사용된다.[23,10] 암호화 과정이 비밀키에 의존하므로 공격자는 비용이 많이 드는 보안 시스템 자체에 대한 공격이나 암호화 알고리즘의 분석보다는 난수 발생기를 분석하여 비밀키를 알아내려고 한다.[23]

이상적인 난수 생성기(truly random bit generator)는 난수를 생성하기위해 물리적인 방법을 이용하므로 난수 생성이 느리고 난수를 생성하는데 많은 비용이 든다. 이런 문제점을 해결하기 위해서 초기에 주어진 난수값을 이용해서 난수 값들을 기계적으로 생성하는 의사 난수 생성기가 고안되었다.[19] 난수는 암호학에 기초한 다양한 네트워크 보안 알고리즘에 사용된다.[23,16] 난수를 크게 2가지로 나눌 수 있는데, 실난수(True random)와 의사난수(Pseudo random)로 나눠질 수 있다.[21]

(1) 실난수(True random numbers)

- 비결정적이다.
- 예측 불가능한 어떤 물리적인 소스로부터 획득: 반도체, 방사선 붕괴 등으로부터 전자소음 혹은 열소음 등

(2) 의사난수(Pesudo- random numbers)

- 컴퓨터는 논리적이고 결정적 이므로, 실난수를 산출하지 못함.
- S/W에 기반을 둔 RNG는 최상의 경우에, 의사난수를 생성 가능.
- PRNG은 길이가 짧은 랜덤 비트열(seed)을 길이가 긴 랜덤에 근접한 비트열로 출력하는 알고리즘.

난수 발생기에는 순수 난수발생기(True Random Number Generator)와 의사난수 발생기(Pseudo Random Number Generator)가 있다.[23]

순수 난수 발생기는 사람이 예측 할 수 없는 무질서소스(chaotic sources)로부터 시드를 취하여 난수를 발생한다. 예를 들면 카메라로 사진을 찍어서 디지털 자료로 변환한 후 얻어지는 픽셀 데이터를 가공하거나[23,12] 공중 전파신호를 이용하여 난수를 생성한다.[23,13] 이러한 순수 난수 발생기는 무질서 소스로부터 시드를 수집하기 위해 외부 디바이스가 필요하다는 단점이 있다.

본 프로젝트에서 사용한 의사난수 생성기란 “Pseudo-random”, 보통의 컴퓨터로 만드는 난수는 Pseudo-Random (슈도 랜덤) 난수다. “Pseudo”란, "가짜의", "모조의", "진짜가 아닌"이라는 뜻이다. "의사 난수"에서 "의사(擬似)"는 "Doctor"라는 뜻이 아니라, “비슷하지만 진짜는 아니다"라는 뜻이고, Pseudo 를 "의사"로 번역한다.[8]

의사 난수 발생기는 시드에 의존하기 보다는 수학적 알고리즘이나 연산을 통해 시드를 가공하여 난수를 발생시킨다. 이미 잘 알려진 암호화 알고리즘을 이용하기 때문에 난수 시퀀스가 예측
가능하므로 rand()함수를 사용할 때에는 시드 값을 계속 바꾸어 주어야 한다는 단점이 있다.
의사난수 생성기법의 종류는 RANDU 를 비롯하여 메르센 트위스터, 선형 합동 생성기, 역 합동생성기, LFSR 등 다수의 기법들이 존재한다.

가. RANDU

RANDU는 선형 합동 생성기를 사용한 유사난수 생성기 루틴으로, 1960년대에 IBM 메인프레임에서 널리 사용된 이후 다른 시스템에서도 널리 사용되었다. 이 루틴이 사용하는 유사난수 생성기는 다음과 같이 정의되었다.

$$X_{n+1} = 65539X_n \bmod 2^{31}, \quad \text{단, } X_0\text{은 홀수}$$

RANDU는 역사상 최악으로 설계된 유사난수 생성기 중 하나로 알려져 있다. 조지 마서글리아는 1968년에 선형 합동 생성기로 만들어진 난수들로 이루어진 좌표를 3차원 공간상에 표시하였을 때 유한한 수의 2차원 평면 중 하나 위에 위치하게 된다는 점(마서글리아 효과)을 밝혔는데, 그의 논문에 따르면 나눔수가 2^{31}인 생성기의 최대 평면 수는 1290개지만 RANDU는 15개의 평면상에 모든 점들이 위치하게 된다.[7]

나. 메르센 트위스터

메르센 트위스터(Mersenne Twister)는 1997년에 마츠모토 마코토(松本 眞)와 니시무라 다쿠지(西村 拓士)가 개발한 유사난수 생성기이다.[3,11] 메르센 트위스터는 동일한 저자들이 개발한 TT800 생성기의 개선판으로, 기존 생성기들의 문제점들을 피하면서 매우 질이 좋은 난수를 빠르게 생성할 수 있도록 설계되었다.

메르센 트위스터의 이름은 난수의 반복 주기가 메르센 소수인 데에서 유래했다. 메르센 트위스터는 그 속도와 난수의 품질 때문에 점점 많은 곳에서 채택되고 있으며, 흔히 주기가 219937 − 1인 MT19937을 사용한다. MT19937과 같으나 생성해 내는 난수가 32비트가 아닌 64비트인 MT19937-64도 쓰이며, 2006년에 동일한 저자들이 발표한 SIMD 기반 메르센 트위스터는 MT19937에 비해 대략 두 배 정도 빠른 것으로 알려져 있다.

난수의 품질에도 불구하고, 메르센 트위스터는 암호학적으로 안전한 유사난수 생성기가 아니다. 즉, 난수의 특성(주기, 난수 범위)을 알고 있을 때 유한한 수의 난수(이 경우 624개)만으로 현재 생성기의 상태를 알아 낼 수 있으며, 그 뒤에 나올 난수를 예측해 낼 수 있다. 암호학적으로 안전한 유사난수 생성기를 얻기 위해서는 해시 함수를 사용해야 하지만 난수의 생성 속도가 낮아진다. 또는 블룸 블룸슘(BBS)과 같이 암호학적으로 안전하게 설계된 생성기를 쓸 수도 있다.

다. 선형합동생성기

선형합동생성기(Linear congruential generator, LCG)는 널리 알려진 유사난수 생성기이다. 선형 합동 생성기는 다음 재귀 관계로 정의된 순열 X_i 을 반환한다.

$$X_{n+1} = (aX_x + c) \bmod m$$

따라서 선형합동생성기는 다음과 같은 인자들로 유일하게 결정된다.

0 < m (나눔수) , 0 ≤ a (곱함수) < m , 0 ≤ c (더함수) < m , 0 ≤ X_0 (초기값) < m

선형 합동 생성기의 상태는 바로 이전에 생성된 난수이며, 이 난수는 최대 m가지 경우가 있으므로 난수의 주기는 최대 m임이 자명하다. 하지만 대부분의 경우 이 주기는 훨씬 짧으며, 최대 주기를 갖기 위한 필요충분조건은 다음과 같다.

(1) c와 m이 서로소여야 한다.

(2) a^{-1}이 m의 모든 소인수로 나눠져야 한다.

(3) m이 4의 배수일 경우, a^{-1}도 4의 배수여야 한다.

선형 합동 생성기는 그 인자들과 마지막으로 생성된 난수를 알면 그 뒤에 만들어질 모든 난수를 예측할 수 있기 때문에 암호학적으로 안전한 유사난수 생성기

가 아니다. 또한 선형 합동 생성기가 생성해 내는 난수의 질은 그 인자에 따라 극적으로 달라지며, 인자에 따라서는 적절치 못한 초기 값 때문에 문제가 생기기도 한다. (ex $c = X_0 = 0$인 경우)[4]

라. 역합동 생성기

역합동생성기(inversive congruential generator, ICG)는 비선형적인 유사난수 생성기로, 선형합동생성기의 연속된 난수가 가지고 있는 상관관계를 없애기 위해 합동 곱셈에 대한 역원을 사용하는 알고리즘이다.[21] 역 합동 생성기의 일반식은 다음과 같다.

$$X_{n+1} = (aX_n^{-1} + c) \bmod m$$

여기서 역합동생성기의 각 인자들은 다음과 같은 성질을 만족해야 한다.

(1) 나눔수 m은 소수여야 한다.

(2) 곱함수 a와 더함수 c는 0보다 크고 나눔수보다 작아야 한다.

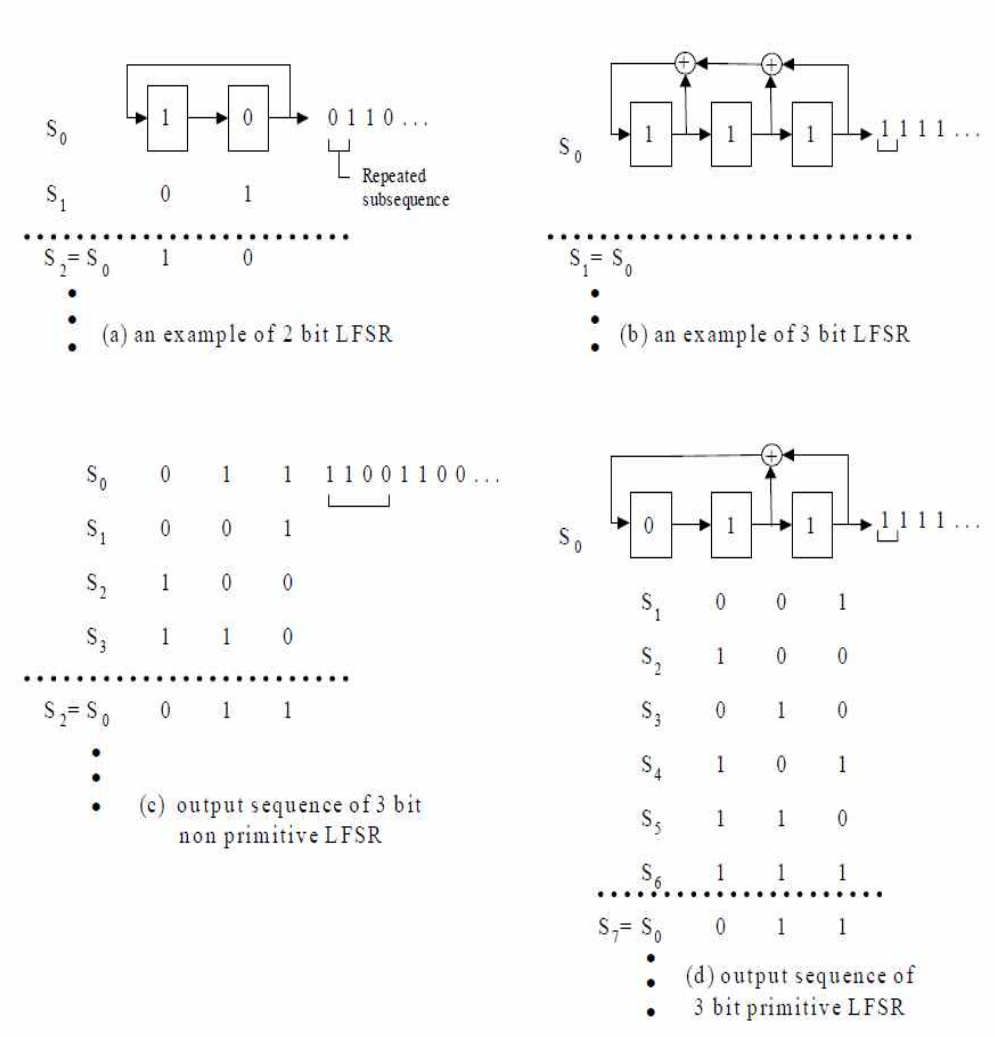
(3) 초기값 X_0은 0보다 크거나 같고 나눔수보다 작아야 한다.

합동 곱셈에 대한 역원 $X_n^{-1} \bmod m$은 유클리드 호제법의 확장으로 계산할 수 있다. 또는 페르마의 소정리에 따라 $X_n^{m-2} \bmod m$을 대신 계산할 수 있다. 이 역원은 나눔수가 소수이기 때문에 X_n이 0이 아니면 항상 유일하게 존재하며, X_n이 0인 경우 후자와 같이 0을 역원으로 쓸 수 있다.[5]

마. LFSR(Linear Feedback Shift Register)

(1) LFSR

LFSR은 수학적으로 분석이 가능한 이진 순열을 효율적으로 발생할 수 있는 장치로 유한체 위에 정의된 선형점화식수열(Linear Recurring Sequence)로 모델링할 수 있으며, 이러한 수열의 특성은 점화식에 의하여 유도되는 특성다항식(Characteristic Polynomial)에 의하여 결정된다.[25]



[그림 8. LFSR의 예]

기본적인 LFSR은 클락 이외에 다른 입력이 없고, D 플립플롭과 XOR로만 구성되어 있으며 출력값들은 일정 길이를 갖고 반복되는 특징을 갖는다. n개의 플립플롭을 갖는 이진 카운터가 0,1, ..., 2n-1 까지의 상태를 계속해서 반복하는 것처럼, n개의 플립플롭을 갖는 LFSR은 최대 2n-1개의 상태를 갖게 된다. [그림8]의 (a)는 단지 2개의 상태(01, 10)를 갖는 것을 알 수 있다. 단 초기값이 00이거나 11인 경우에는 상태변화가 없음을 쉽게 알 수 있다. 마찬가지로 [그림8]의 (b)는 초기상태가 111이거나 000인 경우에는 상태의 변화 없이 항상 일정한 값만을 내

보내는 것을 알 수 있다. 그러나 똑같은 회로에 초기값이 011인 경우에는 [그림 8]의 (c)와 같이 4개의 상태를 갖고 반복됨을 알 수 있다.

[그림8]의 (d)는 길이가 3인 신호를 생성해내며 7개의 반복되는 상태를 갖는 경우를 보여주고 있다. 이때에도 초기값이 모두 0인 경우에는 상태가 변하지 않고 항상 0만을 출력함을 알 수 있다. 초기값이 0이 아닌 경우에 2n-1개의 길이를 갖는 신호를 생성해 내는 회로를 최대길이 (maximal length)를 갖는 쉬프트 레지스터라 부른다. LFSR이 최대 길이를 갖는 시퀀스를 생성하기 위해서는 LFSR의 동작을 수학적으로 표현하여야 하며 적절한 LFSR의 구조 및 LFSR의 초기값의 선택이 BIST를 구성하는데 있어 중요한 요소이다.[17,1] 유한개(k)의 상태를 갖는 회로는 또한 k번의 클락이 인가될 때마다 같은 출력이 반복되게 된다.[17]

(2) 특성다항식

스트림암호에서 사용되는 열쇠 이진수열은 주어진 선형점화식에 의해 생성되는 무한이진 수열들을 원소로 구성된 어떤 다항식의 해 공간 안에서 하나를 열쇠로 선택하여 평문이진 수열을 암호화한다. 따라서 고유다항식은 열쇠의 비도를 결정하는 데 상당한 역할을 한다고 할 수 있다.[22]

임의의 이진 숫자들의 수열 $a_0, a_1, a_2 \cdots, a_m, \cdots$은 발생함수 G($x$)라 불리는 다음과 같은 다항식으로 표시될 수 있다.

$$G(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_m x^m \cdots$$

이때 수열 $a_m = a_0, a_1, a_2, \cdots$ 가 LFSR에 의해 생성되는 출력값이라고 가정하면, 이 수열은 다음과 같이 표시된다.

$$G(x) = \sum_{m=0}^{\infty} a_m x^n$$

LFSR의 초기값을 $a_1, a_2, \cdots, a_{n-1}, a_n$이라 가정하면

$$G(x) = \sum_{m=0}^{INF} \sum_{i=0}^{n} c_i a_{m-i} x^m = \sum_{i=1}^{n} c_i x^i \sum_{m=0}^{\infty} a_{m-i} x^{m-i} = \sum_{i=1}^{n} c_i x^i [a_{-i} x^{-i} + \ldots + a_{-1} x^{-1} + G(x)]$$

따라서 G(x)는 다음과 같이 표현된다.

$$G(x) = \frac{\sum_{i=1}^{n} c_i x^i [a_{-i} x^{-i} + \ldots + a_{-1} x^{-1}]}{1 + \sum_{i=1}^{n} c_i x^i}$$

그러므로 G(x)는 LFSR의 초기값인 $a_1, a_2, \cdots, a_n$ 과 되먹임 상수 $c_1, c_2, \cdots$, cn 의 함수이다. G(x)의 분모는 다음과 같이 표시되며 이를 특성 다항식 P(x)라 부른다.

$$P(x) = 1 + c_1 x + c_2 x^2 + \ldots + c_n x^n$$

특성 다항식은 LFSR의 구조에 따라 결정되며, [그림11]의 특성 다항식 $P(x) = 1 + x + x^2 + x^3$

이며, (d)의 특성 다항식 $P(x) = 1 + x + x^2 + x^3$이다. LFSR이 출력할 수 있는 반복되지 않는 가장 긴 수열의 길이는 플립플롭의 개수를 n이라 할 때 2n-1이며, 이러한 수열을 발생시킬 수 있는 특성다항식을 원시다항식(primitive polynomial)이라 부른다.[17]

(3) 원시다항식

GF(q) 상의 원시다항식은 스크램블러, 에러정정 부호 및 복호기, 난수 발생기 그리고 스트림암호 등 여러 분야에 걸쳐 많이 사용되고 있다.

스트림암호 체계에서 원시다항식을 특성다항식으로 하는 Linear Feedback Shift Register가 많이 사용되고 있다.[26,15]

원시다항식을 찾는 알고리즘은 일반적으로 두 가지 방법으로 분류된다[26,14]

첫 번째 방법은 임의로 주어진 다항식이 원시다항식인가 아닌가를 판정하는 알고리즘을 이용하는 것이고, 둘째 방법은 알고 있는 한 개의 원시다항식으로부터 새로운 원시다항식을 얻는 알고리즘을 이용하는 것이다. 아래의 방식은 첫 번째 방법인 임의로 주어진 다항식이 원시다항식인지 아닌지 판정하는 방법이다.

체 F_2 위의 n 차 원시다항식 $f(x)$ 에 의하여 생성되는 영이 아닌 수열 $\{s_t\} \in \Omega(f(x))$ 을 주기가 $2^n - 1$ 인 최대주기수열(Maximal length sequence) 또는 PN수열(Pseudo-noise sequence)이라 한다.[22]

원시다항식은 임의의 소수 p 와 양의 정수 n 에 대하여 갈로아체 F_{p^n} 에서

$$F_{p^n}^* = F_{p^n} - \{0\} = <\alpha> = \{1, \alpha, \alpha^2, \cdots, \alpha^{p^n-2}\}, \quad \alpha^{p^n-1} = 1$$

인 원소 $\alpha \in F_{p^n}$ 을 체 F_{p^n} 의 원시원소(Primitive element)라 하고 $\alpha \in F_{p^n}$ 의 체 F_p 에서의 최소다항식

$$p(x) = \min.poly._{F_p}\alpha\,,\ p(x) \in F_p[x]$$

을 F_p 위에서의 원시다항식(Primitive polynomial)이라 한다.

① 일반적으로 체 K가 체 F의 확대체일 때 원소 $\alpha \in K$가 체 F위에서 대수적(원소)이면 즉,

$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \in F[x]$ 에 대하여

$$p(\alpha) = a_0 + a_1\alpha + a_2\alpha + \cdots + a_n\alpha^n = 0$$

이면 다음 세 조건을 만족하는 유일한 다항식 $p(x) \in F[x]$ 을 체 F 위에서의 α 의 최소다항식(Minimum polynomial) 또는 기약다항식(Irreducible polynomial)이라 하고 이를

$p(x) = \min.poly._F\alpha$ 또는 $p(x) = irr(\alpha, F)$ 로 나타낸다.

- $\deg(p(x)) \geq 1$ 이고 $p(x)$ 의 최고차항의 계수는 1이다.
- $p(\alpha) = 0$
- 다항식 $f(x) \in F[x]$ 에 대하여 $F[x]$ 에서 $f(\alpha) = 0 \Leftrightarrow p(x) \mid f(x)$

② 위의 다항식 $p(x)$ 의 차를 α 의 F위에서의 차라 하고 이것을 $[\alpha : F] = \deg(p(x))$ 로 나타낸다.

③ 위의 설명에 의하면 최소다항식 또는 기약다항식 중 특별한 경우가 바로 원시다항식이다.

*** 원시다항식의 좋은 예 ***

(다항식 $f(x) = 1 + x + x^4$ 체 F_2 위에서의 4차 원시다항식)

$$\alpha^4 + \alpha + 1 = 0,\ \alpha^4 = \alpha + 1$$

$$F_{2^4} - \{0\} = <\alpha> = \{1, \alpha, \alpha^2, \cdots, \alpha^{14}\},\ \alpha^{15} = 1$$

$\alpha^0 = 1$ $\alpha^1 = \alpha$ $\alpha^2 = \alpha^2$ $\alpha^3 = \alpha^3$

$\alpha^4 = \alpha + 1$ $\alpha^5 = \alpha^2 + \alpha$ $\alpha^6 = \alpha^3 + \alpha^2$

$\alpha^7 = \alpha^4 + \alpha^3 = \alpha + 1 + \alpha^3$

$\alpha^8 = \alpha(\alpha^7) = \alpha(\alpha^3 + \alpha + 1) = \alpha^4 + \alpha^2 + \alpha = \alpha + 1 + \alpha^2 + \alpha = \alpha^2 + 1$

$\alpha^9 = \alpha(\alpha^8) = \alpha(\alpha^2 + 1) = \alpha^3 + \alpha$

$\alpha^{10} = \alpha(\alpha^9) = \alpha(\alpha^3 + \alpha) = \alpha^4 + \alpha^2 = \alpha^2 + \alpha + 1$

$\alpha^{11} = \alpha(\alpha^{10}) = \alpha(\alpha^2 + \alpha + 1) = \alpha^3 + \alpha^2 + \alpha$

$\alpha^{12} = \alpha(\alpha^{11}) = \alpha(\alpha^3 + \alpha^2 + \alpha) = \alpha^4 + \alpha^3 + \alpha^2 = \alpha^3 + \alpha^2 + \alpha + 1$

$\alpha^{13} = \alpha(\alpha^{12}) = \alpha(\alpha^3 + \alpha^2 + \alpha + 1) = \alpha^4 + \alpha^3 + \alpha^2 + \alpha = \alpha^3 + \alpha^2 + 1$

$\alpha^{14} = \alpha(\alpha^{13}) = \alpha(\alpha^3 + \alpha^2 + 1) = \alpha^4 + \alpha^3 + \alpha = \alpha^3 + 1$

$\alpha^{15} = \alpha(\alpha^{14}) = \alpha(\alpha^3 + 1) = \alpha^4 + \alpha = 1$

이므로 정리하면

$\alpha^0 = 0 \cdot \alpha^3 + 0 \cdot \alpha^2 + 0 \cdot \alpha + 1$	(0, 0, 0, 1)
$\alpha^1 = 0 \cdot \alpha^3 + 0 \cdot \alpha^2 + 1 \cdot \alpha + 0$	(0, 0, 1, 0)
$\alpha^2 = 0 \cdot \alpha^3 + 1 \cdot \alpha^2 + 0 \cdot \alpha + 0$	(0, 1, 0, 0)
$\alpha^3 = 1 \cdot \alpha^3 + 0 \cdot \alpha^2 + 0 \cdot \alpha + 0$	(1, 0, 0, 0)
$\alpha^4 = 0 \cdot \alpha^3 + 0 \cdot \alpha^2 + 1 \cdot \alpha + 1$	(0, 0, 1, 1)
$\alpha^5 = 0 \cdot \alpha^3 + 1 \cdot \alpha^2 + 1 \cdot \alpha + 0$	(0, 1, 1, 0)
$\alpha^6 = 1 \cdot \alpha^3 + 1 \cdot \alpha^2 + 0 \cdot \alpha + 0$	(1, 1, 0, 0)
$\alpha^7 = 1 \cdot \alpha^3 + 0 \cdot \alpha^2 + 1 \cdot \alpha + 1$	(1, 0, 1, 1)
$\alpha^8 = 0 \cdot \alpha^3 + 1 \cdot \alpha^2 + 0 \cdot \alpha + 1$	(0, 1, 0, 1)
$\alpha^9 = 1 \cdot \alpha^3 + 0 \cdot \alpha^2 + 1 \cdot \alpha + 0$	(1, 0, 1, 0)
$\alpha^{10} = 0 \cdot \alpha^3 + 1 \cdot \alpha^2 + 1 \cdot \alpha + 1$	(0, 1, 1, 1)
$\alpha^{11} = 1 \cdot \alpha^3 + 1 \cdot \alpha^2 + 1 \cdot \alpha + 0$	(1, 1, 1, 0)
$\alpha^{12} = 1 \cdot \alpha^3 + 1 \cdot \alpha^2 + 1 \cdot \alpha + 1$	(1, 1, 1, 1)
$\alpha^{13} = 1 \cdot \alpha^3 + 1 \cdot \alpha^2 + 0 \cdot \alpha + 1$	(1, 1, 0, 1)
$\alpha^{14} = 1 \cdot \alpha^3 + 0 \cdot \alpha^2 + 0 \cdot \alpha + 1$	(1, 0, 0, 1)
$\alpha^{15} = 0 \cdot \alpha^3 + 0 \cdot \alpha^2 + 0 \cdot \alpha + 1$	(0, 0, 0, 1)

이다. 따라서 다음 수열은 체 F_2 위에서 원시다항식 $f(x)$을 고유다항식으로 하는 주기가 15인 최대주기 수열이다.[22]

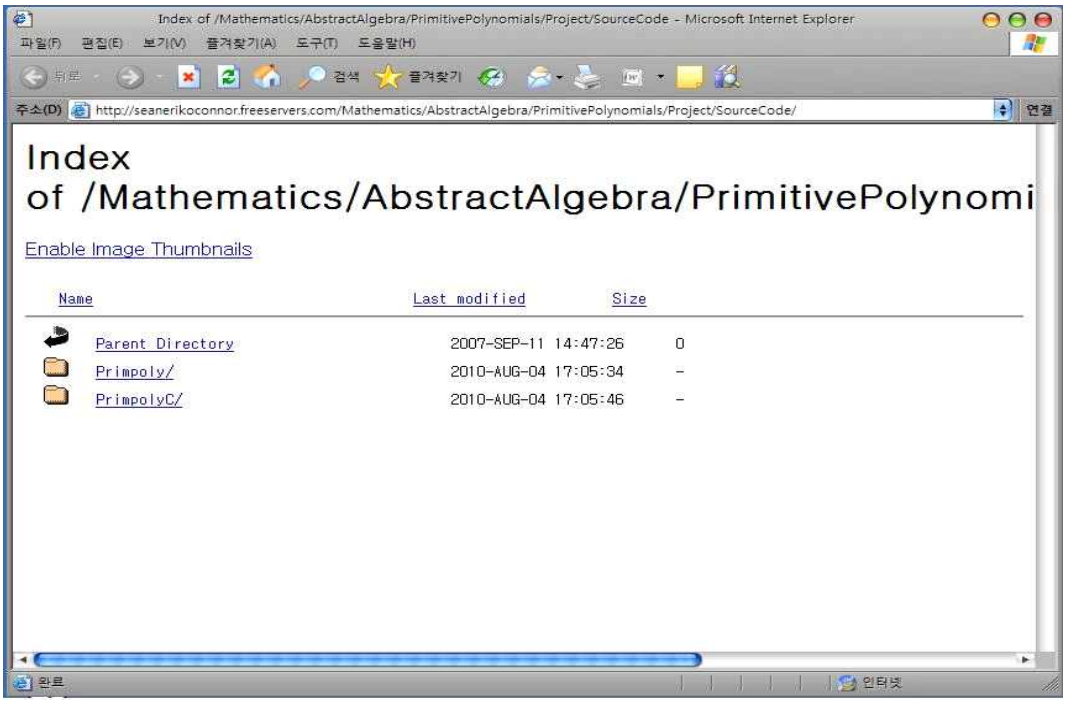
0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0

2.3 LFSR구현 및 검정체제 구축

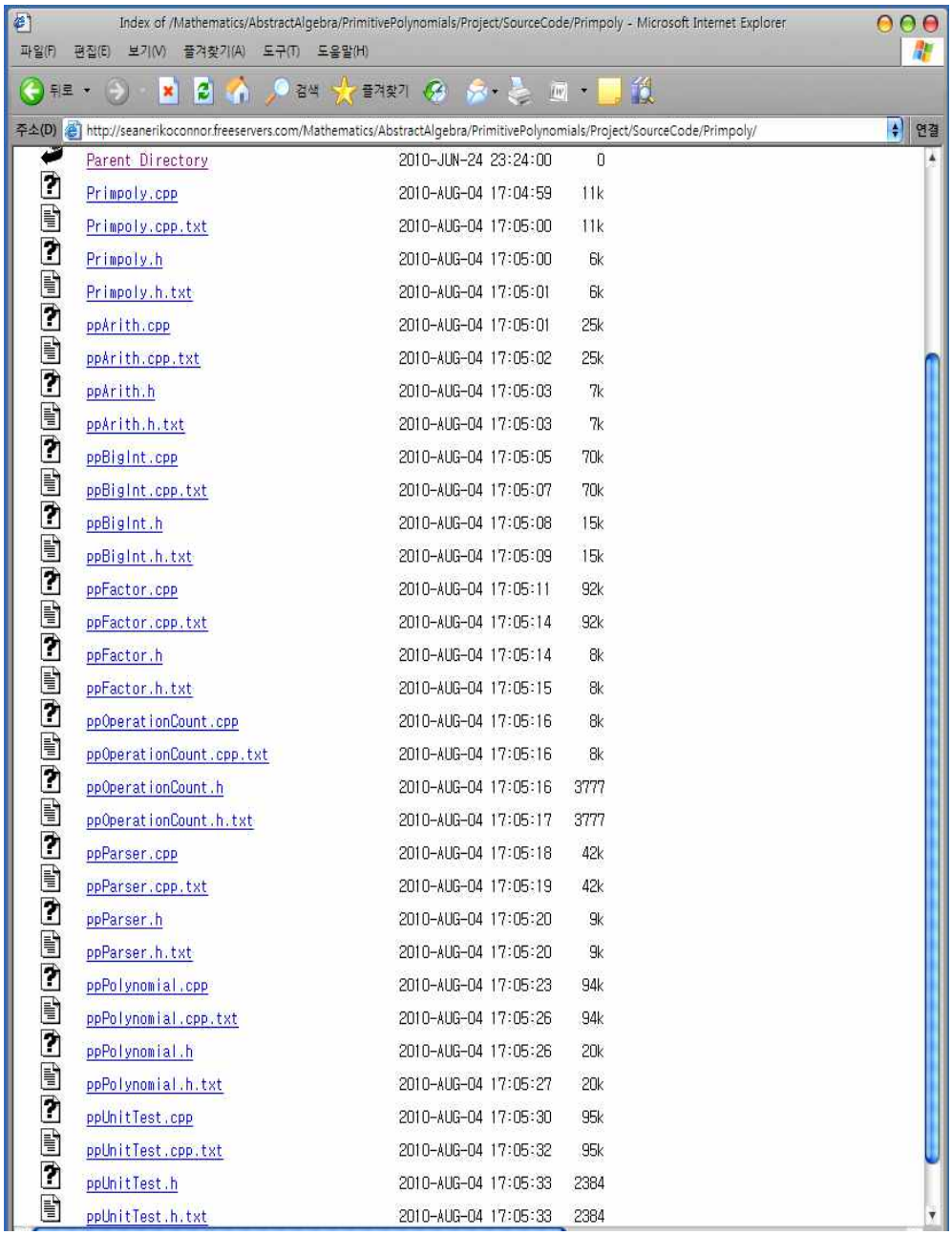
가. LFSR 프로그램 구현

(1) 원시다항식 생성

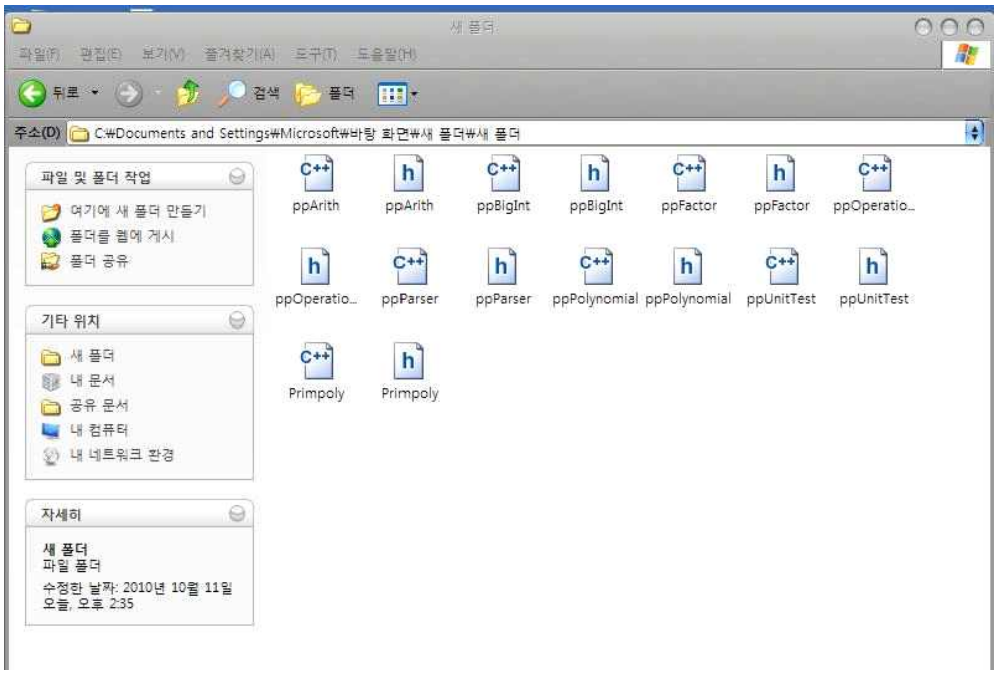
- LFSR을 구현하기 전에 특성다항식으로 사용 할 원시다항식을 생성해야 한다. 이때 사용한 프로그램이 아래와 같다.[9]
  (프로그램이 원시다항식 512차까지 생성가능. 하지만 시간상 512차까지는 출력하지 못함.)
  아래의 프로그램은 웹 서핑으로 찾아낸 것.
- 기존의 프로그램에서 이론을 바탕으로 설계한 원시다항식 생성에서는 31 비트(차)까지만 가능, 기존의 프로그램과 웹에서 찾은 프로그램의 출력값을 비교 후 일치함을 확인 하고 웹에서 찾은 프로그램에서 필요한 부분 수정 후 졸업 작품에 응용
- 수정한 소스파일 ppPolynomial.cpp 에서 출력소스를 바꿔서 필요한 output생성
  fout문을 추가해서 .txt파일을 만들어 낸다. 파일명은 ppdirXXX.txt로 생성한다.
  여기서 XXX는 X비트(n차) 로 입력한다.
- 웹에서 찾은 프로그램은 cygwin에서 호환이 되기 때문에 cygwin으로 컴파일 후 실행.



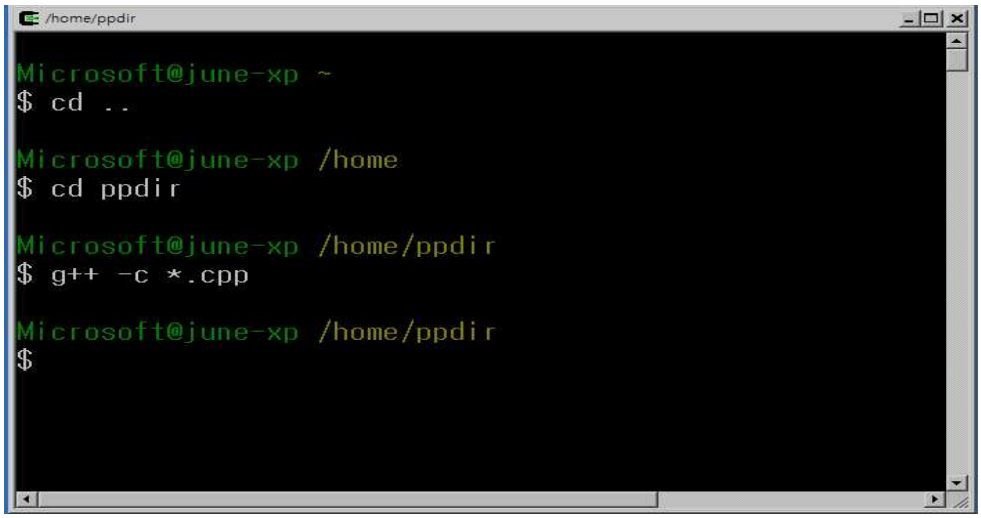
[그림 9. 소스코드 메인 페이지]



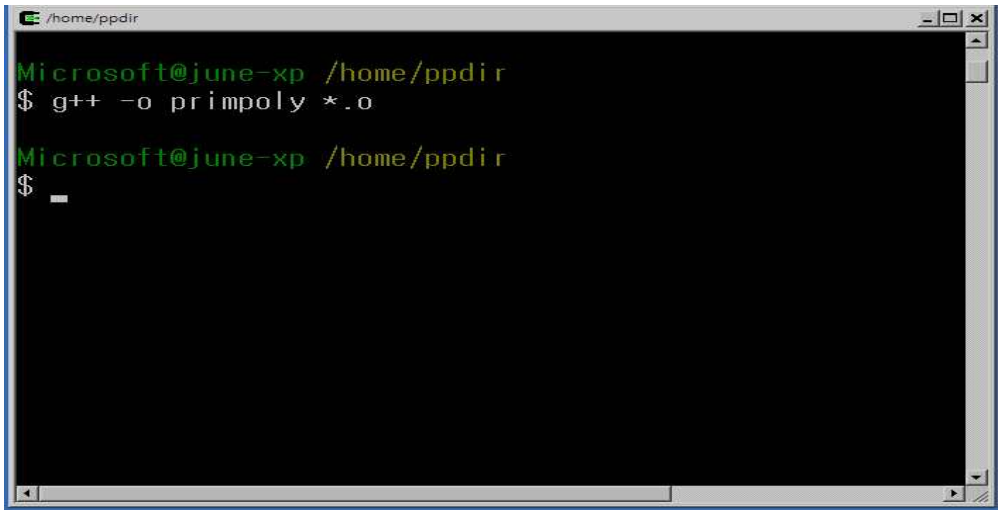
[그림 10. 메인페이지 -> Primpoly/ -> Source - *.cpp 와 *.h 모두 다운]



[그림 11. Source - 다운 후 ppPolynomial.cpp소스 수정]



[그림 12. 모든 *.cpp를 컴파일하여 *.o를 생성]



[그림 13. 컴파일한모든 *.o와 *.h 를 결합 후 실행프로그램 primpoly.exe를 생성]

- primpoly실행

  현재 디렉토리에서 ./primpoly -a 2 xxx

여기서 xxx는 LFSR의 이용할 비트(차) 수로 512도 입력할 수 있으나 원시다항식 생성에 장시간 소요될 것임 (그러나 몇 년이 소요되는 불가능한 정도는 아님)

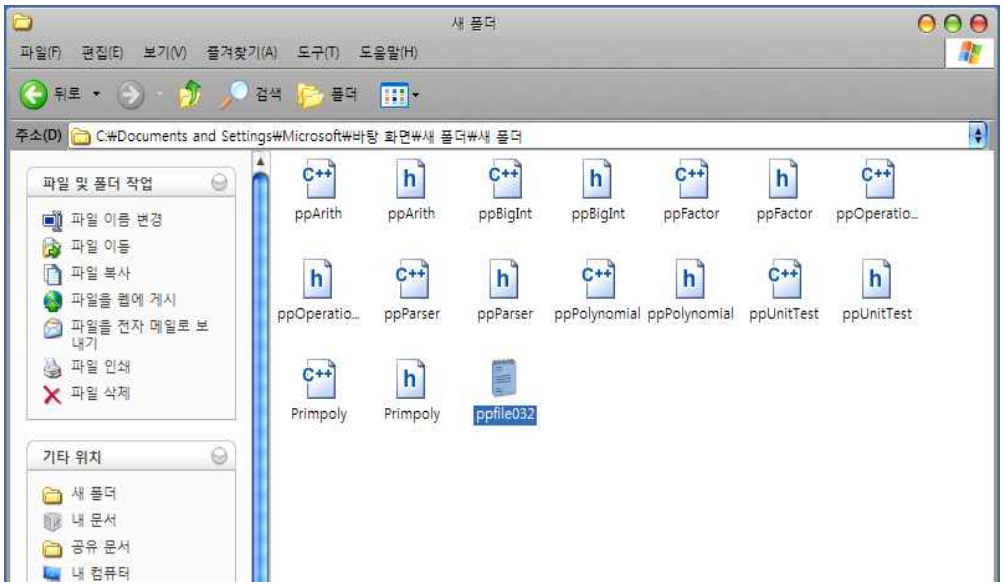
[그림 14. 원시다항식 32비트(32차) 출력]

- primpoly출력

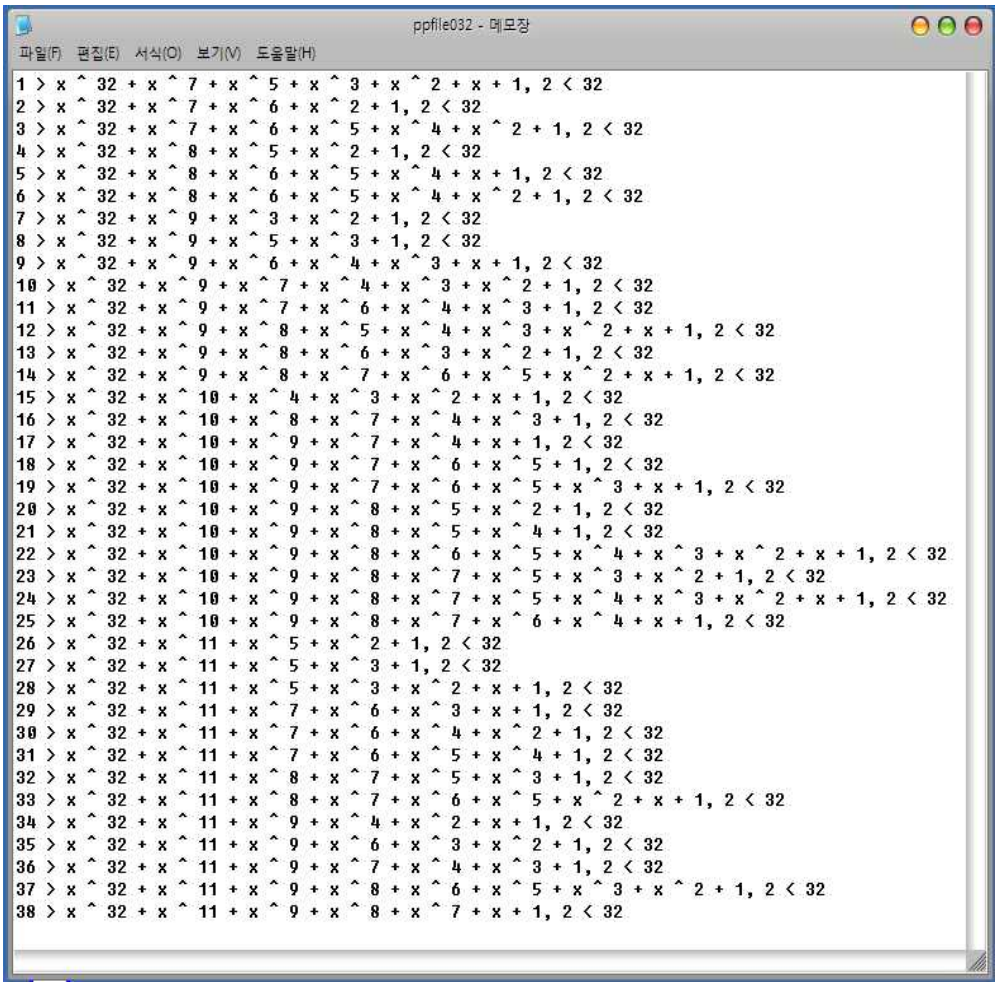
원시다항식의 크기가 클수록(TAP의 비트수가 증가) 작업시간이 길어지며 프로그램 실행시 출력은 화면 출력과 파일 출력(ppfilexxx.txt)에서 차이가 있음.

```
/home/ppdir

Primpoly Version 10.5 - A Program for Computing Primitive Polynomials.
Copyright (C) 1999-2010 by Sean Erik O'Connor.  All Rights Reserved.

Primpoly comes with ABSOLUTELY NO WARRANTY; for details see the
GNU General Public License.  This is free software, and you are welcome
to redistribute it under certain conditions; see the GNU General Public License
for details.

Self-check passes...

>> File Name of primitive polynomials : ppfile032.txt
no = 1 pp = x ^ 32 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 + x + 1, 2> 32
no = 2 pp = x ^ 32 + x ^ 7 + x ^ 6 + x ^ 2 + 1, 2> 32
no = 3 pp = x ^ 32 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 2 + 1, 2> 32
no = 4 pp = x ^ 32 + x ^ 8 + x ^ 5 + x ^ 2 + 1, 2> 32
no = 5 pp = x ^ 32 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x + 1, 2> 32
no = 6 pp = x ^ 32 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 2 + 1, 2> 32
no = 7 pp = x ^ 32 + x ^ 9 + x ^ 3 + x ^ 2 + 1, 2> 32
no = 8 pp = x ^ 32 + x ^ 9 + x ^ 5 + x ^ 3 + 1, 2> 32
no = 9 pp = x ^ 32 + x ^ 9 + x ^ 6 + x ^ 4 + x ^ 3 + x + 1, 2> 32
no = 10 pp = x ^ 32 + x ^ 9 + x ^ 7 + x ^ 4 + x ^ 3 + x ^ 2 + 1, 2> 32
no = 11 pp = x ^ 32 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 4 + x ^ 3 + 1, 2> 32
no = 12 pp = x ^ 32 + x ^ 9 + x ^ 8 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1, 2>
32
no = 13 pp = x ^ 32 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 3 + x ^ 2 + 1, 2> 32
no = 14 pp = x ^ 32 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 2 + x + 1, 2>
32
no = 15 pp = x ^ 32 + x ^ 10 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1, 2> 32
no = 16 pp = x ^ 32 + x ^ 10 + x ^ 8 + x ^ 7 + x ^ 4 + x ^ 3 + 1, 2> 32
no = 17 pp = x ^ 32 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 4 + x + 1, 2> 32
```

[그림 15. 32차 원시다항식 출력]



[그림 16. ppfile032.txt 생성]



ppfile032 - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
1 > x ^ 32 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 + x + 1, 2 < 32
2 > x ^ 32 + x ^ 7 + x ^ 6 + x ^ 2 + 1, 2 < 32
3 > x ^ 32 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 2 + 1, 2 < 32
4 > x ^ 32 + x ^ 8 + x ^ 5 + x ^ 2 + 1, 2 < 32
5 > x ^ 32 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x + 1, 2 < 32
6 > x ^ 32 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 2 + 1, 2 < 32
7 > x ^ 32 + x ^ 9 + x ^ 3 + x ^ 2 + 1, 2 < 32
8 > x ^ 32 + x ^ 9 + x ^ 5 + x ^ 3 + 1, 2 < 32
9 > x ^ 32 + x ^ 9 + x ^ 6 + x ^ 4 + x ^ 3 + x + 1, 2 < 32
10 > x ^ 32 + x ^ 9 + x ^ 7 + x ^ 4 + x ^ 3 + x ^ 2 + 1, 2 < 32
11 > x ^ 32 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 4 + x ^ 3 + 1, 2 < 32
12 > x ^ 32 + x ^ 9 + x ^ 8 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1, 2 < 32
13 > x ^ 32 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 3 + x ^ 2 + 1, 2 < 32
14 > x ^ 32 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 2 + x + 1, 2 < 32
15 > x ^ 32 + x ^ 10 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1, 2 < 32
16 > x ^ 32 + x ^ 10 + x ^ 8 + x ^ 7 + x ^ 4 + x ^ 3 + 1, 2 < 32
17 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 4 + x + 1, 2 < 32
18 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 5 + 1, 2 < 32
19 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 3 + x + 1, 2 < 32
20 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 5 + x ^ 2 + 1, 2 < 32
21 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 5 + x ^ 4 + 1, 2 < 32
22 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1, 2 < 32
23 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 + 1, 2 < 32
24 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1, 2 < 32
25 > x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 4 + x + 1, 2 < 32
26 > x ^ 32 + x ^ 11 + x ^ 5 + x ^ 2 + 1, 2 < 32
27 > x ^ 32 + x ^ 11 + x ^ 5 + x ^ 3 + 1, 2 < 32
28 > x ^ 32 + x ^ 11 + x ^ 5 + x ^ 3 + x ^ 2 + x + 1, 2 < 32
29 > x ^ 32 + x ^ 11 + x ^ 7 + x ^ 6 + x ^ 3 + x + 1, 2 < 32
30 > x ^ 32 + x ^ 11 + x ^ 7 + x ^ 6 + x ^ 4 + x ^ 2 + 1, 2 < 32
31 > x ^ 32 + x ^ 11 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + 1, 2 < 32
32 > x ^ 32 + x ^ 11 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 3 + 1, 2 < 32
33 > x ^ 32 + x ^ 11 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 2 + x + 1, 2 < 32
34 > x ^ 32 + x ^ 11 + x ^ 9 + x ^ 4 + x ^ 2 + x + 1, 2 < 32
35 > x ^ 32 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 3 + x ^ 2 + 1, 2 < 32
36 > x ^ 32 + x ^ 11 + x ^ 9 + x ^ 7 + x ^ 4 + x ^ 3 + 1, 2 < 32
37 > x ^ 32 + x ^ 11 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 3 + x ^ 2 + 1, 2 < 32
38 > x ^ 32 + x ^ 11 + x ^ 9 + x ^ 8 + x ^ 7 + x + 1, 2 < 32
```

[그림 17. ppfile032.txt 파일 내용]

(2) LFSR을 이용한 수열 생성

- LFSR 수열 생성기 실행동작.

첫째. 원하는 비트(n차)입력.

둘째. 출력된 원시다항식에서 원하는 원시다항식 택일.

셋째. 생성할 난수 수 입력.

넷째. 초기치 입력.

[그림 18. 프로그램 초기화면]

[그림 19. 32비트(32차) 입력]

```
>> LFSR의 비트 수를 입력하세요 (4 - 512) : 32

.. 원시 다항식의 파일명칭은 ppfile032.txt 입니다.

.. 원시 다항식은 아래와 같습니다.
1 : x ^ 32 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 + x + 1
2 : x ^ 32 + x ^ 7 + x ^ 6 + x ^ 2 + 1
3 : x ^ 32 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 2 + 1
4 : x ^ 32 + x ^ 8 + x ^ 5 + x ^ 2 + 1
5 : x ^ 32 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x + 1
6 : x ^ 32 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 2 + 1
7 : x ^ 32 + x ^ 9 + x ^ 3 + x ^ 2 + 1
8 : x ^ 32 + x ^ 9 + x ^ 5 + x ^ 3 + 1
9 : x ^ 32 + x ^ 9 + x ^ 6 + x ^ 4 + x ^ 3 + x + 1
10 : x ^ 32 + x ^ 9 + x ^ 7 + x ^ 4 + x ^ 3 + x ^ 2 + 1
11 : x ^ 32 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 4 + x ^ 3 + 1
12 : x ^ 32 + x ^ 9 + x ^ 8 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1
13 : x ^ 32 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 3 + x ^ 2 + 1
14 : x ^ 32 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 2 + x + 1
15 : x ^ 32 + x ^ 10 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1
16 : x ^ 32 + x ^ 10 + x ^ 8 + x ^ 7 + x ^ 4 + x ^ 3 + 1
17 : x ^ 32 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 4 + x + 1
18 : x ^ 32 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 5 + 1
19 : x ^ 32 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 3 + x + 1
20 : x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 5 + x ^ 2 + 1
21 : x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 5 + x ^ 4 + 1
22 : x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + x
 + 1
23 : x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 + 1
24 : x ^ 32 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + x
```

[그림 20. 32비트(32차) 입력 후 화면에 출력되는 원시다항식]

```
79 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 5 + x + 1
80 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 2 +
x + 1
81 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 3 + 1
82 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 4 + x + 1
83 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 +
x ^ 2 + 1
84 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 3 + x ^ 2 + 1
85 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 6 + x ^ 5 + x ^ 3 + 1
86 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + 1
87 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 7 + x ^ 2 + x + 1
88 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 +
x + 1
89 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 6 + x ^ 2 + 1
90 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 4 + x ^ 3 + 1
91 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1
92 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 2 + 1
93 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 4 + 1
94 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 7 + x ^ 4 + x ^ 3 + x ^ 2 + 1
95 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 +
x + 1
96 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 6 + x ^ 3 + 1
97 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 6 + x ^ 5 + x ^ 3 + x + 1
98 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 7 + x ^ 6 + x ^ 3 + x ^ 2 + 1
99 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 3 + 1
100 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 8 + x ^ 6 + x ^ 3 + x ^ 2 + 1

>> 원시 다항식을 선택하십시오 ( 선택 : 해당번호, 통과 : 0 ) _
```

[그림 21. 화면에 출력된 32비트(32차) 원시다항식을 선택하라는 메시지]

```
79 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 5 + x + 1
80 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 2 +
x + 1
81 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 3 + 1
82 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 4 + x + 1
83 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 +
x ^ 2 + 1
84 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 3 + x ^ 2 + 1
85 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 6 + x ^ 5 + x ^ 3 + 1
86 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + 1
87 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 7 + x ^ 2 + x + 1
88 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 +
x + 1
89 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 6 + x ^ 2 + 1
90 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 4 + x ^ 3 + 1
91 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1
92 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 2 + 1
93 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 4 + 1
94 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 7 + x ^ 4 + x ^ 3 + x ^ 2 + 1
95 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 +
x + 1
96 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 6 + x ^ 3 + 1
97 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 6 + x ^ 5 + x ^ 3 + x + 1
98 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 7 + x ^ 6 + x ^ 3 + x ^ 2 + 1
99 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 3 + 1
100 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 8 + x ^ 6 + x ^ 3 + x ^ 2 + 1

>> 원시 다항식을 선택하십시오 ( 선택 : 해당번호, 통과 : 0 ) 89
```

[그림 22. 32비트(32차) 원시다항식 중 해당번호 원시다항식 입력]

- ex) 89번째 원시다항식 선택

```
83 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 +
x ^ 2 + 1
84 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 3 + x ^ 2 + 1
85 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 6 + x ^ 5 + x ^ 3 + 1
86 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + 1
87 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 7 + x ^ 2 + x + 1
88 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 +
x + 1
89 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 6 + x ^ 2 + 1
90 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 4 + x ^ 3 + 1
91 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1
92 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 2 + 1
93 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 4 + 1
94 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 7 + x ^ 4 + x ^ 3 + x ^ 2 + 1
95 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 +
x + 1
96 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 6 + x ^ 3 + 1
97 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 6 + x ^ 5 + x ^ 3 + x + 1
98 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 7 + x ^ 6 + x ^ 3 + x ^ 2 + 1
99 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 3 + 1
100 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 8 + x ^ 6 + x ^ 3 + x ^ 2 + 1

>> 원시 다항식을 선택하십시오 ( 선택 : 해당번호, 통과 : 0 ) 89

.. 선택한 원시 다항식은 x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 6 + x ^ 2 + 1입니
다.
.. 원시 다항식에 의거 설정된 탭은 아래와 같습니다.
0100_0101_0011_0000_0000_0000_0000_0001
.. LFSR의 주기는 2^32 - 1입니다.

>> 생성할 난수의 갯수를 입력하세요 ? 10000_
```

[그림 23. 생성할 난수의 갯수 입력]

```
84 : x ^ 32 + x ^ 12 + x ^ 10 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 3 + x ^ 2 + 1
85 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 6 + x ^ 5 + x ^ 3 + 1
86 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 + 1
87 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 7 + x ^ 2 + x + 1
88 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 4 + x ^ 3 + x ^ 2 +
x + 1
89 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 6 + x ^ 2 + 1
90 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 7 + x ^ 6 + x ^ 4 + x ^ 3 + 1
91 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 4 + x ^ 3 + x ^ 2 + x + 1
92 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 2 + 1
93 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 6 + x ^ 4 + 1
94 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 7 + x ^ 4 + x ^ 3 + x ^ 2 + 1
95 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 9 + x ^ 8 + x ^ 7 + x ^ 5 + x ^ 3 + x ^ 2 +
x + 1
96 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 6 + x ^ 3 + 1
97 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 6 + x ^ 5 + x ^ 3 + x + 1
98 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 7 + x ^ 6 + x ^ 3 + x ^ 2 + 1
99 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 7 + x ^ 6 + x ^ 5 + x ^ 3 + 1
100 : x ^ 32 + x ^ 12 + x ^ 11 + x ^ 10 + x ^ 8 + x ^ 6 + x ^ 3 + x ^ 2 + 1

>> 원시 다항식을 선택하십시오 ( 선택 : 해당번호, 통과 : 0 ) 89

.. 선택한 원시 다항식은 x ^ 32 + x ^ 12 + x ^ 11 + x ^ 8 + x ^ 6 + x ^ 2 + 1입니
다.
.. 원시 다항식에 의거 설정된 탭은 아래와 같습니다.
0100_0101_0011_0000_0000_0000_0000_0001
.. LFSR의 주기는 2^32 - 1입니다.

>> 생성할 난수의 갯수를 입력하세요 ? 10000

>> LFSR의 초기치를 입력하세요 40
```

[그림 24. 초기치 입력]

```
1111001100100011010101100100000000011001000110100001001001 10 생성 비트 8460
0000101111101111110000100111010011111011110111100100010001010 생성 비트 8520
1100011000010011110001101011110101101000100000100100010010 10 생성 비트 8580
1100100010111001100001101100110010101001100010110101111010 01 생성 비트 8640
0010111101100100010010000101101101000110011110010011000101 00 생성 비트 8700
1000000101100011101011111010001000101110000101110000011010 00 생성 비트 8760
1101011100110001010000010100000010001011010110001000010001 01 생성 비트 8820
1011100101010001011111001100001010000100100110010001110000 01 생성 비트 8880
1010001110100010011011111001110001010000110011011111111001 1 생성 비트 8940
0011011000111001110100011010100101000100111010000101110000 00 생성 비트 9000
1001011101001100011110111110010101111011110101010110010110 00 생성 비트 9060
1111110101111011011000000111110001100000111010000011011010 11 생성 비트 9120
1011010111010100110101101001010110100100000011010111001110 11 생성 비트 9180
1110001001110100000111011110100110100111001010111111001010 11 생성 비트 9240
1111010010111011110001101001011000000001010011001010111001 00 생성 비트 9300
1101000100010010001010010101100100010111001000101011100100 11 생성 비트 9360
0101000000110001101101001110000110000100101001101110110000 00 생성 비트 9420
1101100110000011101000110111111111111110000001001011101100 11 생성 비트 9480
1010011000010101110111011001100000011101011111111110110111 01 생성 비트 9540
1101000100011011011000011111111001101101100010011101001001 01 생성 비트 9600
0111111101111101010110111000011010101100000110110100101110 11 생성 비트 9660
0001011000011000100101000010010100101011100010001100011110 00 생성 비트 9720
1010101010000110011101101100111100100101011111111110010011 0 생성 비트 9780
1110000111111100111010101001000110100100110100011101110000 01 생성 비트 9840
1100101000001000111011000000011011100000111011101001010011 001 생성 비트 9900
0001110011111111010101011111101001110011011101010111011000 1000 생성 비트 9960
01111001011111000111010100101010101001 11
.. 생성한 LFSR의 수열은 총 10000 비트입니다.

>> LFSR에 의한 난수 생성작업이 모두 끝났습니다!!
계속하려면 아무 키나 누르십시오 . . .
```

[그림 25. 수열 생성]

ppfile032	2010-09-14 오후 4:...	텍스트 문서	9KB
rndfile032	2010-10-04 오후 1:...	텍스트 문서	11KB

[그림 26. rndfileXXX생성]

```
16 181   [16<---비트수,  181<--- output 비트수]
0100111010000001  <--- LFSR의 TAP
0100111010000001  <--- LFSR 초기치
10000001011100100000010100000111011100101010111101100100011 1
100111111100011100010010001111000011111000111101110101100011110
0100001101110011101001111010010111111000101001101111011111011
1   <--- output 순서대로 총 181비트 출력
```

[그림 27. rndfileXXX.txt]

- rndfile.txt 읽는 법 32비트(32차) rndfile.txt 파일은 출력 수열이 너무 크기 때문에 16비트(16차)rndifle.txt파일로 예로 제시.

나. 생성된 수열의 통계적 검정

- 암호기술이 사용되는 대부분 경우에 난수가 사용된다. 이때 생성되는 난수는 충분한 크기와 충분히 작은 확률로 선택되어야 한다. 예를 들면, DES의 키 공간은 256의 크기를 갖는다. 비밀키 k를 이상적인 난수 생성기로 선택하였다면, 공격자는 올바른 비밀키k를 추측하기 위해서는 평균 255번의 탐색이 필요하다. 그러나 16비트의 랜덤 비밀정보 s를 생성하고, 이 s를 공개함수 f를 이용하여 56비트로 확장하여 비밀키 k를 생성한다면, 공격자는 215번의 탐색으로 비밀키 k를 찾아 낼 수 있다. 전통적으로 일련의 난수 생성에 있어서의 관심 사항은 명확한 통계적 관념상 그 수의 생성 순서가 임의적이어야 한다는 데에 있어 왔다. 다음 두 가지 기준이 수

의 순서가 임의성 증명에 사용된다.

* 균일 분포(Uniform distribution) : 순서상으로 수의 분포가 균일해야함; 즉, 각 수의 출현 빈도가 거의 동일해야 함.

* 독립성(Independence) : 순서 상 어떤 값도 다른 값으로부터 추론될 수 없음. 일련의 수가 균일 분포와 같은 어떤 특정 분포를 갖고 있는지를 판정할 수 있는 명확한 시험 방법은 존재하지 않는다. 일련의 수에 대한 독립성 증명 방법보다는 독립성이 존재하지 않음을 보여주기 위해 사용될 수 있는 여러 가지 시험 방법이 있다.[19]

  - 통계적 검정의 주된 원리는 그 수열이 가지고 있는 각 항들 사이의 독립성을 해치는 상호관련성이나 일양 분포를 따르지 않는 빈도의 편향성 등과 같은 통계적 약점이 존재하는 가를 확인하는 방법이다.[19]
  - 이번 졸업 작품에는 여러 통계법들 중 일반적인 통계법인 도수검정, 도수-m검정, Poker검정, Runs검정, 런의 수 검정, 런의 길이검정, 계열 검정, 자기상관 검정을 사용하였다.
  - 각각의 프로그램들은 생성 된 rndfileXXX.txt파일 안에 들어있는 수열을 읽어 들여와서 난수성을 검사한다.
  - 현 통계 프로그램은 참고문헌 [25][19]를 적극 참고하였다.
  - 검정프로그램은 중첩・비중첩식 도수 검정(각 1개 프로그램), 포커 검정(m : 1-8까지), 런 검정(1-8까지 블록의 런 길이 및 런 수 검정 병행) 계열(d=1) 및 자기상관(d=2-8) 검정 5개 프로그램임.

(1) 도수 검정법

이진수열 $x_1, x_2, \cdots, x_n$ 에 대해 0의 개수를 n_0, 1의 개수를 n_1 이라고 하면 $n_0 + n_1 = n$ 을 만족한다. 도수 검정법은 수열이 난수성을 만족하는 경우, 0과 1의 개수가 같다는 가정에 근거한 검정방법이다. 귀무가설은 '이진 수열이 랜덤하다' 이다. 귀무가설 하에서 한 비트가 0이 될 확률은 1/2, 1이 될 확률도 1/2이다. 귀무가설 하에서 0이 발생하는 기대도수 E 0 는 n/2, 1이 발생하는 기대도수 E 1 도 n/2다. 0이 발생하는 실제 관측도수를 O 0 , 1이 발생하는 실제 관측도수를 O 1이라고 하자.

X^2 적합도 검정을 적용해 보면 검정통계량 X^2 값은 다음과 같다.

$$x^2 = \sum_{i=0}^{l} \frac{(O_i - E_i)^2}{E_i} = \sum_{i=0}^{l} \frac{2\left(O_i - \frac{n}{2}\right)^2}{n} = \frac{n}{2}\left[(\frac{O_0 - O_1}{2})^2 - (\frac{O_0 - O_1}{2})^2\right] = \frac{(n_0 - n_1)^2}{2}$$

위 통계량의 분포는 n 이 클 때 자유도 1인 x^2 분포를 따르며 유의수준 α에 대한 기각
역은 $x^2 > x^2$ (1,α) 이다.

(2) 도수-m 검정법 (중첩방식 & 비중첩방식)

이진수열 x_1 , x_2 ,⋯, x_n 을 m 비트 단위로 나누었을 때, $\left[\frac{n}{m}\right]$개의 (x_{km+1} , x_{km+2} ,⋯ , x_{km+m}), k = 0, 1, 2, ⋯, n - 1 들이 m 차원에서 균등하게 분포되어 있는지를 검정하는 방법이다. 이 때 n 개의 m 비트 블록들은 2^m 개 중의 하나가된다.
(여기서 [x]는 x보다 작지 않은 최소 정수를 의미한다.)

(0, 0, ⋯, 0, 0)인 블록의 개수를 n (0)
(0, 0, ⋯, 0, 1)인 블록의 개수를 n (1)
……
(1, 1, ⋯, 1, 1)인 블록의 개수를 n (2^{m-1})

이라 하자. n 비트 이진수열이 완전한 랜덤 수열이면 위의 2^m 가지 형태 블록의 각각의
확률은$\left[\frac{1}{2^m}\right]$이므로 n개 중에서 위의 형태가 나올 평균 개수는$\left[\frac{n}{2^m}\right]$ 이다. 그러므로 통계량

$$T = \sum_{i=0}^{2^m-1} \frac{\left(n(i) - \frac{n}{2^m}\right)^2}{\frac{n}{2^m}} = \frac{2^m}{n} \sum_{i=0}^{2^m-1} n(i)^2 - n$$

은 근사적으로 자유도가 2^m - 1 인 x^2-분포를 따른다. 주어진 이진수열이 랜덤하면,
T 의 값이 작은 값이 될 것이고, 랜덤하지 않다면 T 의 값이 클 것이다. 따라서 유의수준 α에 대한 기각역은 t > x 2 (2^m - 1 ,a) 이다.

도수 검정법과 도수-m도수 검정은 도수-1과 도수는 같기 때문에 한 프로그램으로 만들었다.

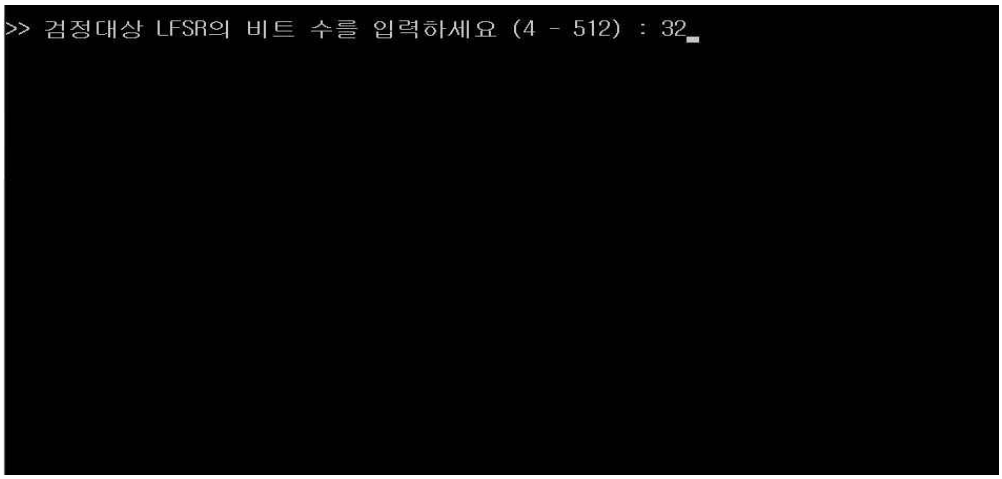
*도수-m 프로그램은 수열을 겹치면서 확인하는 방법과 겹치지 않는 방법 2가지로 만들었다.**

*(각 블록의 실측값 - 각 블록의 기댓값)의 제곱 / 각 블록의 기댓값을 합산.

- 겹치는 도수-m

(도수 m값을 1부터 8 까지 사용하였다.)

*[실행 동작 비트수(n차) 입력(rndfileXXX.txt)]



[그림 28. 겹치는 도수-m 실행 후 입력메시지 화면출력]

도수 (m - 1) 검정 결과

2진법	빈도	2진법	빈도
0	5011	1	4989

생성된 난수의 중첩 1-도수별 기대 값 = 5000.000

생성된 난수의 카이제곱 값 = 0.048

자유도 1의 카이제곱 기준 값 = 3.841

도수 (m - 2) 검정 결과(블록 중첩식)

2진법	빈도	2진법	빈도
00	2502	10	2508
01	2509	11	2480

생성된 난수의 중첩 2-도수별 기대 값 = 2499.750

생성된 난수의 카이제곱 값 = 0.220

자유도 3의 카이제곱 기준 값 = 7.815

도수 (m - 3) 검정 결과(블록 중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
000	1251	010	1289	100	1250	110	1219
001	1251	011	1220	101	1258	111	1260

생성된 난수의 중첩 3-도수별 기대 값 = 1249.750

생성된 난수의 카이제곱 값 = 2.839

자유도 7의 카이제곱 기준 값 = 14.067

도수 (m - 4) 검정 결과(블록 중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
0000	629	0100	650	1000	621	1100	600
0001	622	0101	639	1001	629	1101	619
0010	634	0110	607	1010	655	1110	612
0011	617	0111	612	1011	603	1111	648

생성된 난수의 중첩 4-도수별 기대 값 = 624.813

생성된 난수의 카이제곱 값 = 6.816

자유도 15의 카이제곱 기준 값 = 24.996

[그림 29. 겹치는 도수-m 결과출력1]

도수 (m - 5) 검정 결과(블록 중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
00000	324	01000	329	10000	304	11000	292
00001	305	01001	321	10001	317	11001	308
00010	307	01010	340	10010	327	11010	315
00011	315	01011	299	10011	302	11011	304
00100	333	01100	307	10100	317	11100	293
00101	301	01101	300	10101	338	11101	319
00110	301	01110	292	10110	306	11110	320
00111	315	01111	320	10111	297	11111	328

생성된 난수의 중첩 5-도수별 기대 값 = 312.375

생성된 난수의 카이제곱 값 = 17.497

자유도 31의 카이제곱 기준 값 = 44.985

도수 (m - 6) 검정 결과(블록 중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
000000	166	010000	154	100000	157	110000	150
000001	158	010001	175	100001	147	110001	142
000010	153	010010	169	100010	154	110010	158
000011	152	010011	152	100011	163	110011	150
000100	160	010100	157	100100	173	110100	160
000101	147	010101	183	100101	154	110101	155
000110	152	010110	146	100110	149	110110	160
000111	163	010111	153	100111	152	110111	144
001000	170	011000	150	101000	159	111000	142
001001	163	011001	157	101001	158	111001	151
001010	175	011010	142	101010	165	111010	173
001011	126	011011	158	101011	173	111011	146
001100	140	011100	134	101100	167	111100	159
001101	161	011101	158	101101	139	111101	161
001110	152	011110	171	101110	140	111110	149
001111	163	011111	149	101111	157	111111	179

생성된 난수의 중첩 6-도수별 기대 값 = 156.172

생성된 난수의 카이제곱 값 = 49.581

자유도 63의 카이제곱 기준 값 = 82.529

[그림 30. 겹치는 도수-m 결과출력2]

도수 (m - 7) 검정 결과(블록 중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
0000000	81	0100000	77	1000000	84	1100000	80
0000001	85	0100001	77	1000001	73	1100001	70
0000010	72	0100010	85	1000010	81	1100010	69
0000011	86	0100011	90	1000011	66	1100011	73
0000100	81	0100100	86	1000100	79	1100100	87
0000101	72	0100101	83	1000101	75	1100101	71
0000110	75	0100110	76	1000110	77	1100110	73
0000111	77	0100111	75	1000111	86	1100111	77
0001000	77	0101000	81	1001000	93	1101000	78
0001001	83	0101001	76	1001001	80	1101001	82
0001010	82	0101010	94	1001010	93	1101010	71
0001011	65	0101011	89	1001011	61	1101011	84
0001100	61	0101100	86	1001100	79	1101100	81
0001101	91	0101101	60	1001101	70	1101101	79
0001110	82	0101110	74	1001110	70	1101110	66
0001111	81	0101111	79	1001111	82	1101111	78
0010000	81	0110000	76	1010000	73	1110000	74
0010001	89	0110001	74	1010001	86	1110001	68
0010010	91	0110010	83	1010010	78	1110010	75
0010011	72	0110011	74	1010011	80	1110011	76
0010100	71	0110100	72	1010100	86	1110100	88
0010101	104	0110101	70	1010101	79	1110101	85
0010110	67	0110110	87	1010110	79	1110110	73
0010111	59	0110111	71	1010111	94	1110111	73
0011000	75	0111000	67	1011000	75	1111000	75
0011000	75	0111000	67	1011000	75	1111000	75
0011001	65	0111001	67	1011001	92	1111001	84
0011010	76	0111010	89	1011010	66	1111010	84
0011011	85	0111011	69	1011011	73	1111011	77
0011100	65	0111100	86	1011100	69	1111100	73
0011101	87	0111101	85	1011101	71	1111101	76
0011110	91	0111110	71	1011110	80	1111110	78
0011111	72	0111111	78	1011111	77	1111111	101

생성된 난수의 중첩 7-도수별 기대 값 = 78.078

생성된 난수의 카이제곱 값 = 112.416

자유도 127의 카이제곱 기준 값 = 154.302

[그림 31. 겹치는 도수-m 결과출력3]

도수 (m - 8) 검정 결과(블록 중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
00000000	44	01000000	39	10000000	36	11000000	45
00000001	37	01000001	38	10000001	48	11000001	35
00000010	37	01000010	42	10000010	35	11000010	39
00000011	48	01000011	35	10000011	38	11000011	31
00000100	35	01000100	38	10000100	46	11000100	41
00000101	37	01000101	47	10000101	35	11000101	28
00000110	44	01000110	44	10000110	31	11000110	33
00000111	42	01000111	46	10000111	35	11000111	40
00001000	33	01001000	47	10001000	44	11001000	46
00001001	48	01001001	39	10001001	35	11001001	41
00001010	38	01001010	51	10001010	44	11001010	42
00001011	34	01001011	32	10001011	31	11001011	29
00001100	35	01001100	44	10001100	26	11001100	35
00001101	40	01001101	32	10001101	51	11001101	38
00001110	36	01001110	34	10001110	46	11001110	36
00001111	41	01001111	41	10001111	40	11001111	41
00010000	35	01010000	41	10010000	46	11010000	32
00010001	42	01010001	40	10010001	47	11010001	46
00010010	42	01010010	36	10010010	49	11010010	42
00010011	41	01010011	40	10010011	31	11010011	40
00010100	32	01010100	49	10010100	39	11010100	37
00010101	50	01010101	45	10010101	54	11010101	34
00010110	36	01010110	40	10010110	31	11010110	39
00010111	29	01010111	49	10010111	30	11010111	45
00011000	29	01011000	42	10011000	46	11011000	33

중간 생략 . . .

00101010	51	01101010	31	10101010	43	11101010	40
00101011	53	01101011	39	10101011	36	11101011	45
00101100	35	01101100	41	10101100	51	11101100	40
00101101	32	01101101	46	10101101	28	11101101	33
00101110	33	01101110	33	10101110	41	11101110	33
00101111	26	01101111	38	10101111	53	11101111	40
00110000	36	01110000	36	10110000	40	11110000	38
00110001	39	01110001	31	10110001	35	11110001	37
00110010	37	01110010	38	10110010	46	11110010	37
00110011	28	01110011	29	10110011	46	11110011	47
00110100	39	01110100	45	10110100	33	11110100	43
00110101	37	01110101	44	10110101	33	11110101	41
00110110	46	01110110	36	10110110	41	11110110	37
00110111	39	01110111	33	10110111	32	11110111	40
00111000	37	01111000	40	10111000	30	11111000	35
00111001	28	01111001	46	10111001	39	11111001	38
00111010	48	01111010	47	10111010	41	11111010	37
00111011	39	01111011	38	10111011	30	11111011	39
00111100	52	01111100	38	10111100	34	11111100	35
00111101	39	01111101	33	10111101	46	11111101	43
00111110	29	01111110	31	10111110	42	11111110	47
00111111	43	01111111	47	10111111	35	11111111	54

생성된 난수의 중첩 8-도수별 기대 값 = 39.035

생성된 난수의 카이제곱 값 = 231.142

자유도 255의 카이제곱 기준 값 = 292.175

32비트 에서 중첩도수-m 검정을 총 [100%]통과하였습니다.

[그림 32. 겹치는 도수-m 결과출력4]

- 비중첩 도수-m

중첩도수는 블록의 크기만큼 검사를 한 후 바로 한 칸 뒤로 밀려난 뒤 다시 같은 크기의 블록을 연속해서 검사하는 방법이며, 비중첩은 검사한 비트 다음 번째의 비트를 검사하는 방식이다.

앞에서도 언급했지만, 중첩 도수와 비중첩 도수 두 가지를 만든 이유는 논문 자료에서 중첩의 내용과 비중첩의 내용이 있으며, 수열의 크기가 클 때, 비중첩과 중첩의 비중은 그렇게 크지 않기 때문에 두 가지 프로그램을 만들었다.

[그림 33. 비중첩 도수-m 실행 후 입력메시지 화면출력]

```
                    도수 ( m - 1 ) 검정 결과
                    ========================

                  -------------------------------
                  2진법   빈도       2진법   빈도
                  -------------------------------
                   0      5011        1      4989
                  -------------------------------

            생성된 난수의 비중첩 1-도수별 도수별 기대 값  = 5000.000

            생성된 난수의 카이제곱 값     =   0.048

            자유도   1의 카이제곱 기준 값 =   3.841

            도수 ( m - 2 ) 검정 결과(블록 비중첩식)
            ======================================

                  -------------------------------
                  2진법   빈도       2진법   빈도
                  -------------------------------
                   00     1244        10     1256
                   01     1267        11     1233
                  -------------------------------

            생성된 난수의 비중첩 2-도수별 도수별 기대 값  = 1250.000

            생성된 난수의 카이제곱 값     =   0.520

            자유도   3의 카이제곱 기준 값 =   7.815
            도수 ( m - 3 ) 검정 결과(블록 비중첩식)
            ======================================

 ----------------------------------------------------------------
  2진법    빈도    2진법    빈도    2진법    빈도    2진법    빈도
 ----------------------------------------------------------------
 000       394    010      453    100      437    110      402
 001       420    011      404    101      403    111      420
 ----------------------------------------------------------------

            생성된 난수의 비중첩 3-도수별 도수별 기대 값  = 416.625

            생성된 난수의 카이제곱 값     =   6.797

            자유도   7의 카이제곱 기준 값 =  14.067
            도수 ( m - 4 ) 검정 결과(블록 비중첩식)
            ======================================

 ----------------------------------------------------------------
  2진법    빈도    2진법    빈도    2진법    빈도    2진법    빈도
 ----------------------------------------------------------------
 0000      164    0100     185    1000     142    1100     136
 0001      156    0101     149    1001     146    1101     156
 0010      154    0110     172    1010     170    1110     157
 0011      143    0111     154    1011     145    1111     171
 ----------------------------------------------------------------

            생성된 난수의 비중첩 4-도수별 도수별 기대 값  = 156.250

            생성된 난수의 카이제곱 값     =  16.800

            자유도  15의 카이제곱 기준 값 =  24.996
```

[그림 34. 비중첩 도수-m 결과출력1]

도수 (m - 5) 검정 결과(블록 비중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
00000	74	01000	63	10000	62	11000	58
00001	60	01001	55	10001	70	11001	60
00010	59	01010	65	10010	67	11010	57
00011	64	01011	45	10011	63	11011	58
00100	64	01100	65	10100	60	11100	60
00101	49	01101	64	10101	88	11101	64
00110	62	01110	64	10110	63	11110	55
00111	62	01111	70	10111	65	11111	65

생성된 난수의 비중첩 5-도수별 도수별 기대 값 = 62.500

생성된 난수의 카이제곱 값 = 26.592

자유도 31의 카이제곱 기준 값 = 44.985

도수 (m - 6) 검정 결과(블록 비중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
000000	26	010000	28	100000	26	110000	28
000001	27	010001	41	100001	25	110001	22
000010	34	010010	33	100010	23	110010	28
000011	21	010011	21	100011	34	110011	24
000100	22	010100	29	100100	28	110100	30
000101	28	010101	28	100101	28	110101	24
000110	16	010110	25	100110	33	110110	26
000111	28	010111	24	100111	28	110111	24
001000	20	011000	19	101000	22	111000	23
001001	33	011001	28	101001	23	111001	24
001010	27	011010	26	101010	25	111010	28
001011	24	011011	24	101011	39	111011	24
001100	26	011100	23	101100	23	111100	31
001101	20	011101	25	101101	20	111101	28
001110	23	011110	28	101110	23	111110	22
001111	23	011111	20	101111	27	111111	33

생성된 난수의 비중첩 6-도수별 도수별 기대 값 = 26.031

생성된 난수의 카이제곱 값 = 51.397

자유도 63의 카이제곱 기준 값 = 82.529

[그림 35. 비중첩 도수-m 결과출력2]

도수 (m - 7) 검정 결과(블록 비중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
0000000	14	0100000	10	1000000	15	1100000	8
0000001	10	0100001	14	1000001	16	1100001	13
0000010	7	0100010	16	1000010	9	1100010	14
0000011	15	0100011	11	1000011	13	1100011	12
0000100	12	0100100	10	1000100	12	1100100	15
0000101	4	0100101	9	1000101	12	1100101	9
0000110	11	0100110	11	1000110	9	1100110	13
0000111	12	0100111	14	1000111	10	1100111	14
0001000	8	0101000	3	1001000	12	1101000	16
0001001	10	0101001	6	1001001	14	1101001	13
0001010	13	0101010	16	1001010	14	1101010	11
0001011	11	0101011	12	1001011	10	1101011	13
0001100	10	0101100	12	1001100	7	1101100	12
0001101	13	0101101	6	1001101	4	1101101	8
0001110	15	0101110	6	1001110	9	1101110	9
0001111	14	0101111	12	1001111	7	1101111	14
0010000	16	0110000	12	1010000	7	1110000	9
0010001	19	0110001	7	1010001	13	1110001	7
0010010	13	0110010	13	1010010	10	1110010	4
0010011	9	0110011	13	1010011	12	1110011	10
0010100	10	0110100	11	1010100	10	1110100	14
0010101	16	0110101	11	1010101	9	1110101	10
0010110	12	0110110	9	1010110	16	1110110	3
0010111	8	0110111	11	1010111	19	1110111	10
0011000	10	0111000	5	1011000	10	1111000	8
0011001	12	0111001	11	1011001	9	1111001	9
0011010	14	0111010	11	1011010	7	1111010	13
0011011	15	0111011	14	1011011	10	1111011	14
0011100	8	0111100	14	1011100	17	1111100	12
0011101	14	0111101	14	1011101	10	1111101	13
0011110	12	0111110	11	1011110	7	1111110	7
0011111	13	0111111	8	1011111	13	1111111	16

생성된 난수의 비중첩 7-도수별 도수별 기대 값 = 11.156

생성된 난수의 카이제곱 값 = 118.218

자유도 127의 카이제곱 기준 값 = 154.302

[그림 36. 비중첩 도수-m 결과출력3]

도수 (m - 8) 검정 결과(블록 비중첩식)

2진법	빈도	2진법	빈도	2진법	빈도	2진법	빈도
00000000	6	01000000	4	10000000	2	11000000	8
00000001	5	01000001	5	10000001	7	11000001	3
00000010	5	01000010	9	10000010	1	11000010	6
00000011	7	01000011	6	10000011	1	11000011	2
00000100	7	01000100	6	10000100	4	11000100	8
00000101	5	01000101	6	10000101	2	11000101	4
00000110	6	01000110	6	10000110	4	11000110	7
00000111	8	01000111	4	10000111	3	11000111	5
00001000	6	01001000	6	10001000	5	11001000	8
00001001	4	01001001	6	10001001	4	11001001	6
00001010	7	01001010	8	10001010	4	11001010	2
00001011	5	01001011	8	10001011	4	11001011	1
00001100	4	01001100	5	10001100	1	11001100	6
00001101	7	01001101	7	10001101	8	11001101	4
00001110	2	01001110	1	10001110	3	11001110	4
00001111	6	01001111	6	10001111	4	11001111	5
00010000	3	01010000	3	10010000	5	11010000	2
00010001	4	01010001	7	10010001	7	11010001	5
00010010	6	01010010	4	10010010	4	11010010	4
00010011	5	01010011	4	10010011	1	11010011	5
00010100	6	01010100	6	10010100	6	11010100	4
00010101	10	01010101	8	10010101	5	11010101	1
00010110	6	01010110	3	10010110	4	11010110	7
00010111	3	01010111	6	10010111	2	11010111	5
00011000	3	01011000	7	10011000	8	11011000	3
00101010	5	01101010	8	10101010	2	11101010	10
00101011	4	01101011	3	10101011	3	11101011	8
00101100	4	01101100	5	10101100	6	11101100	2
00101101	4	01101101	6	10101101	5	11101101	7
00101110	7	01101110	1	10101110	6	11101110	5
00101111	3	01101111	6	10101111	7	11101111	6
00110000	5	01110000	5	10110000	5	11110000	5
00110001	4	01110001	1	10110001	5	11110001	7
00110010	4	01110010	4	10110010	7	11110010	4
00110011	6	01110011	3	10110011	2	11110011	4
00110100	1	01110100	4	10110100	2	11110100	8
00110101	1	01110101	4	10110101	3	11110101	4
00110110	6	01110110	9	10110110	2	11110110	9
00110111	8	01110111	1	10110111	2	11110111	5
00111000	8	01111000	5	10111000	3	11111000	3
00111001	4	01111001	4	10111001	9	11111001	3
00111010	8	01111010	8	10111010	4	11111010	7
00111011	5	01111011	9	10111011	2	11111011	7
00111100	7	01111100	4	10111100	1	11111100	4
00111101	2	01111101	6	10111101	5	11111101	4
00111110	1	01111110	4	10111110	5	11111110	5
00111111	8	01111111	5	10111111	9	11111111	6

생성된 난수의 비중첩 8-도수별 도수별 기대 값 = 4.883

생성된 난수의 카이제곱 값 = 242.582

자유도 255의 카이제곱 기준 값 = 292.175

32비트 에서 비중첩도수-m 검정을 총 [100%]통과하였습니다.

[그림 37. 비중첩 도수-m 결과출력4]

(3) Poker 검정법

도수-m 검정법은 크기 m인 블록을 겹치기 않게 나누어 각 블록의 가능한 패턴이 고른지를 검정하는 방법이다. 반면에 poker 검정법은 크기 m인 블록을 겹치기 않게 나누어 각 블록의 1의 개수만을 고려한 패턴이 고른지를 검정하는 방법이다. Porker 검정법을 hamming weight 검정법이라고도 한다. 도수-m 검정에서 사용한 통계량 보다 단순한 다음의 통계량을 사용하면 공식이 간단하여 쉽게 이용할 수 있다. 그러나 도수-m 검정에서 사용한 통계량보다는 많은 정보의 누수가 발생하므로 정밀성은 도수-m 검정 통계량을 사용한 것보다 떨어진다.

도수-m 검정법과 같이 수열을 크기가 m 인 $\left[\frac{n}{m}\right]$개의 블록으로 나눈다. poker 검정에서는

n (i) (0 ≤i≤m)를 i 개의 1과 m - i 개의 0으로 구성된 블록의 개수라 하자. 수열이 랜덤하면 n (i)의 평균은 $\overline{n(i)}-\binom{m}{i}, 0 \le i \le m$이다. 그러므로

$$T=\sum_{i=0}^{m}\frac{(n(i)-\binom{m}{i}\frac{n}{2^m})^2}{\binom{m}{i}\frac{n}{2^m}}=\frac{2^m}{n}\sum_{i=0}^{m}\frac{(n(i))^2}{\binom{m}{i}}-n$$

는 근사적으로 자유도가 m인 x^2-분포를 따른다.

- (각 블록의 실측값 - 각 블록의 기댓값)의 제곱 / 각 블록의 기댓값을 합산

```
>> 검정대상 LFSR의 비트 수를 입력하세요 (4 - 512) : 32_
```

[그림 38. poker검정 실행 후 입력메시지 화면출력]

```
                    Pocker - 1 검정 결과
                    ====================
                      1의 갯수     빈도
                  -------------------------
                        0          7433
                        1          7567
                  -------------------------
            생성된 Poker 검정의 카이제곱 값     =   1.197

            자유도   1의 카이제곱 기준 값 =   3.841

                    Pocker - 2 검정 결과
                    ====================
                      1의 갯수     빈도
                  -------------------------
                        0          1838
                        1          3757
                        2          1905
                  -------------------------
            생성된 Poker 검정의 카이제곱 값     =   1.223

            자유도   3의 카이제곱 기준 값 =   5.991

                    Pocker - 3 검정 결과
                    ====================
                      1의 갯수     빈도
                  -------------------------
                        0           634
                        1          1824
                        2          1883
                        3           659
                  -------------------------
            생성된 Poker 검정의 카이제곱 값     =   3.401

            자유도   7의 카이제곱 기준 값 =   7.815

                    Pocker - 4 검정 결과
                    ====================
  1의 갯수   빈도        1의 갯수   빈도        1의 갯수   빈도
-------------------------------------------------------------------
     0       238           1        909           2       1392
     3       970           4        241
-------------------------------------------------------------------
            생성된 Poker 검정의 카이제곱 값     =   2.381

            자유도  15의 카이제곱 기준 값 =   9.488
```

[그림 39. poker검정 결과출력1]

```
                    Pocker - 5 검정 결과
                    ====================
 1의 갯수   빈도        1의 갯수   빈도        1의 갯수   빈도
-----------------------------------------------------------------
    0       101           1       479           2       888
    3       927           4       494           5       111
-----------------------------------------------------------------
          생성된 Poker 검정의 카이제곱 값     =   8.050

          자유도  31의 카이제곱 기준 값 =  11.070


                    Pocker - 6 검정 결과
                    ====================
 1의 갯수   빈도        1의 갯수   빈도        1의 갯수   빈도
-----------------------------------------------------------------
    0        47           1       214           2       570
    3       795           4       584           5       248
    6        42
-----------------------------------------------------------------
          생성된 Poker 검정의 카이제곱 값     =   5.079

          자유도  63의 카이제곱 기준 값 =  12.592
                    Pocker - 7 검정 결과
                    ====================
 1의 갯수   빈도        1의 갯수   빈도        1의 갯수   빈도
-----------------------------------------------------------------
    0        27           1       103           2       344
    3       561           4       591           5       379
    6       127           7        10
-----------------------------------------------------------------
          생성된 Poker 검정의 카이제곱 값     =  14.955

          자유도 127의 카이제곱 기준 값 =  14.067


                    Pocker - 8 검정 결과
                    ====================
 1의 갯수   빈도        1의 갯수   빈도        1의 갯수   빈도
-----------------------------------------------------------------
    0        15           1        56           2       189
    3       402           4       506           5       412
    6       229           7        59           8         7
-----------------------------------------------------------------
          생성된 Poker 검정의 카이제곱 값     =  12.485

          자유도 255의 카이제곱 기준 값 =  15.507

       64비트 에서 Pocker 검정을 총 [88%]통과하였습니다.
```

[그림 40. poker검정 결과출력2]

(4) Runs검정

- 런의 수 검정

n 개의 이진 수열 x_1 , x_2 ,… , x_n 에 대해 런(runs)을 고려해 보자. 런이란 연속적으로 같은 값이 나열된 부분 수열로, 1로 구성된 런을 'block', 0으로 구성된 런을 'gap'이라고 한다. 예를 들어 이진수열이 '00111011'일 때 'block'은 '111' '11'이며 'gap'은 '00' '0'으로, block 길이가 2인 것이 1개 3인 것이 1개, gap 길이가 2인 것이 1개, 1인 것이 1개이다. 새로운 런의 시작은 change'라고 하면 위의 예에서 change는 3번 일어났으며 런의 개수는 4이다. 런의 개수는 change 수에다 1을 더하면 된다. 즉 runs의 수 =change의 수 +1이다.

이진수열이 랜덤 하다고 가정하면 change가 일어날 확률은 1/2이며, n개의이진 수열에서 change의 개수는 평균이 (n -1)/2, 분산이 (n -1)/22인 이항분포를 따른다. 런의 수(block의 수 + gap의 수)를 R 이라고 하면 R 의 확률 분포는 change 수의 분포를 이용하여 구할 수 있다.

길이 n인 이진수열을 S (S는 n비트 블록 개수)개라고 하면 런의 개수가 i 인 평균 블록 개수 e_i 는 다음과 같다.

$$e_i = \binom{n-1}{i-1} 2^{l-n} \times S$$

S개의 블록 중 런의 개수가 i 인 블록 개수의 관측 값을 o_i 라고 하고, 검정 통계량을 $Q = \sum_{i=1}^{n} \frac{(o_i - e_i)}{e_i}$ 로 두면, 귀무가설 하에서 Q 는 자유도 n - 1 인 x^2 - 분포를 가지며 유의수준 a 에 대한 기각역은 $q > x_a^2 (n-1)$ 이다.

- 런의 길이검정

런의 수뿐만 아니라 런의 길이를 조사하는 것은 중요하다. x_1 , x_2 , …, x_n 을 길이가 n 인 2진 수열이라 할 때, 0 혹은 1이 연속하여 나타나는 정도가 추정치에 근접한 지를 확인하는 검정 방법이다. 길이가 i 인 run에 대하여 1의 run(block)의 개수와 0의 run(gap)의 개수를 각각 B_i , G_i 라고 하면, 길이가 i 인 block(gap)의 개수의 기대 값 E_i 는 다음과 같다.

$$E_{i=}\begin{cases} \dfrac{\dfrac{n-i-1}{2}+2}{2^{i+1}} & .i=1,2,\cdots .n-1 \\ \dfrac{1}{2^n} & .i=n \end{cases}$$

검정 검정통계량을 다음과 같이 정의하자.

$$x^2=\sum_{i=1}^{k}\frac{(B_i-E_i)^2}{E_i}+\sum_{i=1}^{k}\frac{(G_i-E_i)^2}{E_i}$$

그러면 검정 검정통계량 x^2 는 자유도가 2k -2인 x^2 분포를 따른다.

- 각 런의 (길이별 실측값 - 길이별 기댓값)의 제곱 / 길이 기댓값을 합산하는 경우(런 길이 검정)
- 각 런의 (수별 실측값 - 수별 기댓값)의 제곱 / 수별 기댓값을 합산하는 경우(런 수 검정)
- 런의 수 와 런의 길이 검정은 하나의 프로그램.

```
>> 검정대상 LFSR의 비트 수를 입력하세요 (4 - 512) : 32
```

[그림 41. runs검정 실행 후 입력메시지 화면출력]

블록 - 2 런 검정 결과

1. 런 길이 검정

Block 런 길이	측정 값	Gap 런 길이	측정 값	block/gap 기대 값
1	2494	1	2494	2500.000
2	1248	2	1258	1250.000

생성 난수의 런의 길이 카이제곱 값 = 0.083

자유도 2의 카이제곱 기준 값 = 5.990

2. 런 수 검정

런 수	런 수 측정 값	런 수 기대 값
1	2506	2500.000
2	2494	2500.000

생성된 난수의 런의 수 카이제곱 값 = 0.029

자유도 1의 카이제곱 기준 값 = 3.840

블록 - 3 런 검정 결과

1. 런 길이 검정

Block 런 길이	측정 값	Gap 런 길이	측정 값	block/gap 기대 값
1	2106	1	2065	2083.125
2	792	2	826	833.250
3	433	3	431	416.625

생성 난수의 런의 길이 카이제곱 값 = 3.654

자유도 4의 카이제곱 기준 값 = 9.490

2. 런 수 검정

런 수	런 수 측정 값	런 수 기대 값
1	864	833.333
2	1618	1666.667
3	851	833.333

생성된 난수의 런의 수 카이제곱 값 = 2.924

자유도 2의 카이제곱 기준 값 = 5.990

[그림 42. runs검정 결과출력1]

블록 - 4 런 검정 결과

1. 런 길이 검정

Block 런 길이	측정 값	Gap 런 길이	측정 값	block/gap 기대 값
1	1901	1	1866	1875.000
2	758	2	780	781.250
3	307	3	332	312.500
4	163	4	147	156.250

생성 난수의 런의 길이 카이제곱 값 = 3.250

자유도 6의 카이제곱 기준 값 = 12.590

2. 런 수 검정

런 수	런 수 측정 값	런 수 기대 값
1	310	312.500
2	947	937.500
3	922	937.500
4	321	312.500

생성된 난수의 런의 수 카이제곱 값 = 0.604

자유도 3의 카이제곱 기준 값 = 7.810

블록 - 5 런 검정 결과

1. 런 길이 검정

Block 런 길이	측정 값	Gap 런 길이	측정 값	block/gap 기대 값
1	1750	1	1757	1750.000
2	733	2	736	750.000
3	308	3	318	312.500
4	125	4	128	125.000
5	70	5	63	62.500

생성 난수의 런의 길이 카이제곱 값 = 1.812

자유도 8의 카이제곱 기준 값 = 15.510

2. 런 수 검정

런 수	런 수 측정 값	런 수 기대 값
1	133	125.000
2	510	500.000
3	742	750.000
4	466	500.000
5	149	125.000

생성된 난수의 런의 수 카이제곱 값 = 7.717

자유도 4의 카이제곱 기준 값 = 9.490

[그림 43. runs검정 결과출력2]

블록 - 6 런 검정 결과

1. 런 길이 검정

Block 런 길이	측정 값	Gap 런 길이	측정 값	block/gap 기대 값
1	1706	1	1649	1666.000
2	705	2	722	728.875
3	302	3	333	312.375
4	131	4	121	130.156
5	57	5	53	52.063
6	26	6	28	26.031

생성 난수의 런의 길이 카이제곱 값 = 4.971

자유도 10의 카이제곱 기준 값 = 18.310

2. 런 수 검정

런 수	런 수 측정 값	런 수 기대 값
1	54	52.083
2	279	260.417
3	485	520.833
4	529	520.833
5	264	260.417
6	55	52.083

생성된 난수의 런의 수 카이제곱 값 = 4.203

자유도 5의 카이제곱 기준 값 = 11.070

블록 - 7 런 검정 결과

1. 런 길이 검정

Block 런 길이	측정 값	Gap 런 길이	측정 값	block/gap 기대 값
1	1660	1	1665	1606.500
2	654	2	697	714.000
3	304	3	301	312.375
4	157	4	128	133.875
5	51	5	59	55.781
6	21	6	26	22.313
7	14	7	12	11.156

생성 난수의 런의 길이 카이제곱 값 = 16.321

자유도 12의 카이제곱 기준 값 = 21.030

2. 런 수 검정

런 수	런 수 측정 값	런 수 기대 값
1	26	22.321
2	132	133.929
3	320	334.821
4	438	446.429
5	350	334.821
6	137	133.929
7	25	22.321

생성된 난수의 런의 수 카이제곱 값 = 2.529

자유도 6의 카이제곱 기준 값 = 12.590

[그림 44. runs검정 결과출력3]

블록 - 8 런 검정 결과

1. 런 길이 검정

Block 런 길이	측정 값	Gap 런 길이	측정 값	block/gap 기대 값
1	1605	1	1590	1562.500
2	703	2	704	703.125
3	282	3	321	312.500
4	145	4	133	136.719
5	58	5	53	58.594
6	18	6	26	24.414
7	13	7	8	9.766
8	8	8	5	4.883

생성 난수의 런의 길이 카이제곱 값 = 11.163

자유도 14의 카이제곱 기준 값 = 23.680

2. 런 수 검정

런 수	런 수 측정 값	런 수 기대 값
1	13	9.766
2	58	68.359
3	188	205.078
4	346	341.797
5	361	341.797
6	207	205.078
7	68	68.359
8	9	9.766

생성된 난수의 런의 수 카이제곱 값 = 5.274

자유도 7의 카이제곱 기준 값 = 14.070

32비트 에서 런 검정을 총 [88%]통과하였습니다.

[그림 45. runs검정 결과출력4]

(5) 계열(Serial) 검정법

계열 검정법은 이진수열 $x_1, x_2, \cdots, x_n$ 에서 한 비트가 그 다음 비트로 전이되는 과정이 랜덤한지를 조사하기 위한 검정법이다. 이진수열을 연속된 2 비트 $x_1 x_2, x_2 x_3, x_3 x_4, \cdots, x_{n-1} x_n$ 으로 나누었을 때 “00”의 개수를 n_{00} , “11”의 개수를 n_{11} 이라 하고 전체 n 비트 중 “0”의 개수를 n_0 , ″1″의 개수를 n_1 이라 하자. 그러면 다음의 식을 얻는다.

$$n_{00}+n_{01}=n_0 \text{ 혹은 } n_0-1$$
$$n_{10}+n_{11}=n_1 \text{ 혹은 } n_n-1$$
$$n_{00}+n_{01}+n_{10}+n_{11}=n-1$$
$$n_0+n_1=n$$

위 식에서 -1이 나타내는 이유는 길이 n 개의 부분에서는 단지 n - 1 개의 전이가 일어나기 때문이다. n_{ij} 의 기대치는 $\frac{n-1}{4}$ 이므로 다음의 통계량은 근사적으로 자유도 2인 x ^{2}-분포를 따른다.

$$T=\sum_{i,j=0}^{l}\frac{(n_{ij}-\frac{n-1}{4})^2}{\frac{n-1}{4}}-\sum_{i=0}^{l}\frac{(n_i-\frac{n}{2})^2}{\frac{n}{2}}$$

이것을 다른 식으로 표현하면

$$T=\frac{4}{n-1}(n_{00}^2+n_{01}^2+n_{10}^2+n_{11}^2)-\frac{2}{n}(n_0^2+n_1^2)+1$$

로 간단한 식이 된다. 계열 검정은 계열 상관검정(serial correlation test)으로 알려져 있기도 한다.

일반적으로 길이가 b인 부분 수열을 겹치게 생성하여 2^b 개의 부분 수열에 대한 형태의 균일성을 조사하는 검정 방법이다. 부분 수열 (x_1 , x_2 , ⋯, x_b) 는

$$t=\sum_{i=1}^{b}2^i x_i$$

$0 \le t \le 2^b - 1$로 표현될 수 있으며, n_t 를 임의의 부분 수열에서 발생하는 10진수의 값 t 의 개수라고 하자. 길이가 b 인 부분 수열을 갖는 이진수열의 균일성에 대한 검정 통계량을 다음과 같이 정의한다.

$$\psi_b^2=\frac{2^b}{n-b-1}\sum_{t=b}^{2^b-1}(n_r-\frac{n-b+1}{2^b})^2, \text{ 여기서 } \psi_b^2=0$$

이제 0의 개수를 n_0 , 1의 개수를 n_1이라고 하자. b = 2 라고 했을 때, 즉, 부분 수열

(x_1 , x_2),(x_2 , x_3), ⋯, (x_{n-1}, x_n) 에서 발생하는 형태는 00, 01, 10, 11이며 각각의 개수를 n_{00} , n_{01} , n_{10} , n_{11}이라고 한다. 귀무가설은 '각각의 형태에서 발생하는 개수가 같다'이다. 따라서 검정 통계량 ψ^2_2 (b = 2일 때 ψ^2_2) 는 다음과 같다.

$$\psi_2^2 = \frac{4}{n-1}\sum_{t=0}^{3}(n_t - \frac{n-1}{4})^2$$
$$= \frac{4}{n-1}\left[(n_0 - \frac{n-1}{4})^2 + (n_1 - \frac{n-1}{4})^2 + (n_2 - \frac{n-1}{4})^2 + n_3 - \frac{n-1}{4})^2\right]$$

여기서 부분 수열의 형태는 00, 01, 10, 11이므로 $n_0 = n_{00}$, $n_1 = n_{01}$, $n_2 = n_{10}$, $n_3 = n_{11}$ 이라 두면

$$\psi_2^2 = \frac{4}{n-1}\left[(n_0 - \frac{n-1}{4})^2 + (n_1 - \frac{n-1}{4})^2 + (n_2 - \frac{n-1}{4})^2 + (n_3 - \frac{n-1}{4})^2\right]$$

이다. 또한 다른 통계량 ψ^2_{b-1} 는

$$\psi_{b-1}^2 = \frac{2^{b-1}}{n-(b-1)+1}\sum_{t=0}^{2^{b-1}l-1}(n_1 - \frac{n-(b-1)+1}{2^{b-1}})^2$$

이다. b=2를 대입하면

$$\psi_1^2 = \frac{2}{n}\sum_{i=0}^{l}(n_i - \frac{n}{2})^2$$
$$= \frac{2}{n}\left[(n_0 - \frac{n}{2})^2 + (n_1 - \frac{n}{2})^2\right]$$

이다. 즉, ψ^2_1은 도수 검정 통계량이다. 위의 ψ^2_2 와 ψ_1^2 에 대한 통계량 ψ^2 를 $\psi^2 = \psi_2^2 - \psi_1^2$ 로 두면

$$\psi^2 = \frac{4}{n-1}(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2(n_0^2 + n_1^2)}{n} + 1$$

이다. ψ^2 은$n \geq 21$일 때 자유도가 2인 x^2 분포를 따른다.

(6) 자기상관(Autocorrelation) 검정법

계열검정은 바로 뒤에 있는 비트로 전이되어 가는 과정을 조사한 것이다. 이와 같은 과정을 d 비트 떨어진 비트간의 전이를 생각한 것이 자동상관검정이다. 이제까지 알려진 자동상관검정에서 검정 통계량들은 다음 4가지가 있다.

① (x_1 , x_{d+1}),(x_2 , x_{d+2}), …, (x_{n-d} , x_n) 로 이루어진 2비트들에 대하여 계열검정과 마찬가지로 n_{ij} 를 구하여 검정하는 것이 자동상관검정이다.

② 이진수열 n 비트 x_1 , x_2 , … x_n 으로부터 (x_1 , x_2 , …, x_{n-d})와(x_{d+1} , x_{d+2} , …, x_n)와의 상관관계를 조사하는 통계량으로 다음을 생각하여 보자.

$$A(d) = \sum_{i=l}^{n-d} (1-2x_i)(1-2x_{d+i})$$

이진수열이 랜덤 하면, 즉, 독립이고 같은 분포를 가지면, 통계량 A (d)의 평균은 0이고 분산은 n - d 이다. n - d 가 충분히 클 경우 (예를 들면 n - d ≥ 30)에는 중심 극한 정리에 의하여 $Z = \frac{A(d)}{\sqrt{n-d}}$ 는 근사적으로 표준 정규분포 N (0, 1)을 따른다.[19]

- 자기상관 검정 : 계열 검정의 연속으로 d=2부터 8까지 검정하는 것으로 했음. 논리적 근거는 같은 논문 같은 쪽 중간에 기술됨.

[그림 46. 계열검정 실행 후 입력메시지 화면출력]

계열(serial) 검정 결과

2진법	빈 도	2진법	빈 도	2진법	빈 도
0	5011	00	2502	10	2508
1	4988	01	2509	11	2480

계열 검정의 카이제곱 값 = 2.167

자유도 2의 카이제곱 기준 값 = 7.810

자기상관(d = 2) 검정 결과

2진법	빈 도	2진법	빈 도	2진법	빈 도
0	5011	00	2540	10	2469
1	4987	01	2471	11	2518

자기상관 검정의 카이제곱 값 = 3.433

자유도 2의 카이제곱 기준 값 = 7.810

자기상관(d = 3) 검정 결과

2진법	빈 도	2진법	빈 도	2진법	빈 도
0	5010	00	2520	10	2488
1	4987	01	2490	11	2499

자기상관 검정의 카이제곱 값 = 2.204

자유도 2의 카이제곱 기준 값 = 7.810

자기상관(d = 4) 검정 결과

2진법	빈 도	2진법	빈 도	2진법	빈 도
0	5009	00	2533	10	2474
1	4987	01	2476	11	2513

자기상관 검정의 카이제곱 값 = 2.954

자유도 2의 카이제곱 기준 값 = 7.810

[그림 47. 계열검정 결과출력1]

자기상관(d = 5) 검정 결과

2진법	빈 도	2진법	빈 도	2진법	빈 도
0	5009	00	2491	10	2515
1	4986	01	2518	11	2471

자기상관 검정의 카이제곱 값 = 2.532

자유도 2의 카이제곱 기준 값 = 7.810

자기상관(d = 6) 검정 결과

2진법	빈 도	2진법	빈 도	2진법	빈 도
0	5008	00	2518	10	2487
1	4986	01	2490	11	2499

자기상관 검정의 카이제곱 값 = 2.185

자유도 2의 카이제곱 기준 값 = 7.810

자기상관(d = 7) 검정 결과

2진법	빈 도	2진법	빈 도	2진법	빈 도
0	5008	00	2524	10	2480
1	4985	01	2484	11	2505

자기상관 검정의 카이제곱 값 = 2.444

자유도 2의 카이제곱 기준 값 = 7.810

자기상관(d = 8) 검정 결과

2진법	빈 도	2진법	빈 도	2진법	빈 도
0	5007	00	2531	10	2472
1	4985	01	2476	11	2513

자기상관 검정의 카이제곱 값 = 2.940

자유도 2의 카이제곱 기준 값 = 7.810

32비트 에서 자기상관&계열 검정을 총 [100%]통과하였습니다.

[그림 48. 계열검정 결과출력2]

3. 최종 목표 및 결과의 활용

가. 결 과

스트림암호에서 사용하는 LFSR(Linear Feedback Shift Register)을 이용한 의사난수생성기구현하였다. 이 때 LFSR에서 사용하는 특성다항식을 원시다항식으로 사용하였으며, 이 원시다항식을 찾는 방법은 n비트(차)에 대한 다항식이 원시다항식인지 확인하는 프로그램으로 원시다항식을 찾아내서 LFSR안에서 사용.

원시다항식을 사용한 LFSR은 암호의 비예측성 중 하나인 최대주기 $2^n - 1$ 개의 수열을 .txt로 출력, 31비트 이상의 비트에 대한 최대주기는 .txt로 입력받기에 시간과 용량면에서 제약이 있기 때문에, 31비트이상은 출력하지 않음.

생성된 수열이 난수성이 있는지 없는지 확인하기 위해서 일반적으로 사용하는 일반통계학을 이론으로 프로그램을 5개 구현.

이 프로그램들은 도수검정, 도수-m검정, 포커검정, 런의 수 와 런의 길이검정, 계열검정 과 자기상관검정 으로 나누어져 있다. 도수-m검정이 중복검사와 비중복검사로 나눠서 만들었기 때문에 프로그램이 5개가 된다.

이 모든 프로그램은 논문[19]을 이용하여 프로그램화했었는데, 도수-m프로그램에 대한 논문 내용이 중첩내용과 비중첩에 대한 내용이 있기에 두가지 프로그램을 모두 구현 후 그 두 가지 프로그램도 모두 검정작업에 사용.

수열의 길이가 커지면서 중첩과 비중첩의 비도는 그렇게 크지 않다는 것을 알았다.

모든 통계프로그램을 통과하는 수열, 비트(차), 원시다항식, 초기치 이 4가지를 우리는 NS(Normal Set)라고 정함.

만약 암호문이 난수성 검정을 통과하지 못했다고 해서 그 암호알고리즘이 비난수(Non-Random)적이라고 결론지을 수는 없다. 그 이유는 모집단으로부터 선택된 극히 일분분의 표본(Sample)을 가지고 데이터집합을 구성하여 검정을 실시하기 때문에 비난수성을 입증할 충분한 근거를 제시하는 것이 불가능하기 때문에 [25] NS(Normal Set)라고 정함.

검정시 모든 비트를 다 검정하지 않고 2의 배수인 32비트, 48비트, 64비트, 80비

트, 96비트, 112비트, 128비트, 192비트 총 8개의 비트를 검정. 192비트 이상을 검정하고 싶었지만 컴퓨터 속도가 필요한 연산을 실행하는데, 많은 시간이 지나도 출력물이 나오지 않았다. 2주 이상 출력물을 기다려도 출력되지 않아서 192까지만 검정하기로 정함. PC 보다 빠른 미니급이나 나아가 슈퍼컴을 동원할 경우 에만 그 이상이 원활한 속도로 나올 수 있을 것이다.

완벽하게 다 통과하는 비트 또는 원시다항식, 초기값, 수열을 찾지 못했지만, 고정적인 NS는 찾았다. 현재 결과는 초기치, 비트, 원시다항식, 수열 모두 수열의 안전성에 영향을 미치기 때문에, NS를 사용하여야 안전한 수열이 될 수 있으며 NS자료들은 다음과 같다.

- 통계 프로그램 검사를 통과한 NS(Normal set) List -

비트 (차)	원시다항식	수열	초기치
32	$x^{32} + x^7 + x^6 + x^2 + 1$	5500	3
	$x^{32} + x^7 + x^6 + x^5 + x^4 + x^2 + 1$		
	$x^{32} + x^8 + x^5 + x^2 + 1$		
	$x^{32} + x^8 + x^6 + x^5 + x^4 + x + 1$		
	$x^{32} + x^9 + x^6 + x^4 + x^3 + x + 1$		
	$x^{32} + x^9 + x^8 + x^7 + x^6 + x^5 + x^2 + x + 1$		
	$x^{32} + x^{10} + x^9 + x^7 + x^4 + x + 1$		
	$x^{32} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$		
	$x^{32} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$		
	$x^{32} + x^{11} + x^5 + x^3 + 1$		
	$x^{32} + x^{11} + x^5 + x^3 + x^2 + x + 1$		
	$x^{32} + x^{11} + x^7 + x^6 + x^4 + x^2 + 1$		
	$x^{32} + x^{11} + x^7 + x^6 + x^5 + x^4 + 1$		
	$x^{32} + x^{11} + x^9 + x^7 + x^4 + x^3 + 1$		
	$x^{32} + x^{11} + x^9 + x^7 + x^4 + x^3 + 1$		
	$x^{32} + x^{11} + x^9 + x^7 + x^4 + x^3 + 1$		
	$x^{32} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^3 + x^2 + 1$		
	$x^{32} + x^{11} + x^9 + x^8 + x^7 + x + 1$		
	$x^{32} + x^{11} + x^{10} + x^4 + x^3 + x + 1$		
	$x^{32} + x^{11} + x^{10} + x^4 + x^3 + x^2 + 1$		
	$x^{32} + x^{11} + x^{10} + x^4 + x^3 + x^2 + 1$		
	$x^{32} + x^{11} + x^{10} + x^6 + x^5 + x^4 + 1$		
	$x^{32} + x^{11} + x^{10} + x^7 + x^6 + x^4 + x^3 + x + 1$		
	$x^{32} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + 1$		

[표 1. (32차) NS]

비트 (차)	원시다항식	수열	초기치
48	$x^{48} + x^{8} + x^{7} + x^{5} + x^{2} + x + 1$ $x^{48} + x^{10} + x^{8} + x^{5} + x^{4} + x + 1$ $x^{48} + x^{10} + x^{9} + x^{7} + x^{6} + x^{5} + x^{2} + x + 1$ $x^{48} + x^{11} + x^{8} + x^{2} + 1$ $x^{48} + x^{11} + x^{9} + x^{8} + x^{7} + x^{6} + x^{4} + x^{2} + 1$ $x^{48} + x^{11} + x^{10} + x^{9} + x^{8} + x^{5} + x^{4} + x^{3} + 1$ $x^{48} + x^{12} + x^{7} + x^{3} + x^{2} + x + 1$ $x^{48} + x^{12} + x^{8} + x^{6} + x^{3} + x^{2} + 1$ $x^{48} + x^{12} + x^{8} + x^{6} + x^{4} + x^{3} + x^{2} + x + 1$ $x^{48} + x^{12} + x^{11} + x^{7} + x^{5} + x^{4} + x^{3} + x^{2} + 1$	4000	3

[표 2. (48차) NS]

비트 (차)	원시다항식	수열	초기치
64	$x^{64} + x^7 + x^6 + x^5 + x^4 + x^2 + 1$	15000	3, 4, 5, 12, 26, 27
	$x^{64} + x^9 + x^8 + x^7 + x^6 + x^3 + x^2 + x + 1$		3, 4, 5, 14, 15, 34, 50, 55, 56, 57, 76, 87
	$x^{64} + x^{11} + x^7 + x^6 + x^5 + x^2 + 1$		3, 12, 13, 18, 19, 26, 27, 30, 31, 36, 37
	$x^{64} + x^{12} + x^8 + x^7 + x^6 + x^5 + x^3 + x^2 + 1$		3, 4, 5, 10, 11, 13, 14, 15, 18, 19, 20, 21, 22
	$x^{64} + x^{12} + x^9 + x^7 + x^5 + x^2 + 1$		3, 10, 11, 19, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 50
	$x^{64} + x^{12} + x^9 + x^7 + x^5 + x^3 + x^2 + x + 1$		3, 4, 5, 10, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 23
	$x^{64} + x^{12} + x^{10} + x^9 + x^6 + x^3 + 1$		3, 4, 5, 6, 7, 8, 9, 10, 11
	$x^{64} + x^{12} + x^{11} + x^9 + x^4 + x^3 + x^2 + x + 1$		3, 6, 7, 22, 23, 24, 25, 38, 39, 48, 49
	$x^{64} + x^{12} + x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$		3, 4, 5, 6, 7, 8, 9, 14, 15, 16, 17
	$x^{64} + x^{12} + x^{11} + x^9 + x^8 + x^6 + x^4 + x^2 + 1$		3, 20, 21, 22, 23, 40, 41, 46, 47, 58, 59
	$x^{64} + x^{12} + x^{11} + x^{10} + x^7 + x^5 + x^4 + x^3 + 1$		3, 4, 5, 6, 7, 24, 25, 54, 55, 58, 59
	$x^{64} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^2 + x + 1$		3, 8, 9, 16, 17, 24, 25, 30, 31, 46, 47

[표 3. (64차) NS]

비트 (차)	원시다항식	수열	초기치
80	$x^{80} + x^{13} + x^{10} + x^{9} + x^{8} + x^{5} + x^{4} + x + 1$	10000	3
	$x^{80} + x^{10} + x^{7} + x^{6} + x^{3} + x^{2} + 1$ $x^{80} + x^{12} + x^{10} + x^{3} + x^{2} + x + 1$ $x^{80} + x^{12} + x^{11} + x^{7} + x^{6} + x^{5} + x^{3} + x^{2} + 1$ $x^{80} + x^{12} + x^{11} + x^{8} + x^{6} + x^{5} + x^{3} + x^{2} + 1$ $x^{80} + x^{13} + x^{10} + x^{9} + x^{8} + x^{6} + x^{5} + x^{4} + x^{3} + x + 1$ $x^{80} + x^{13} + x^{11} + x^{9} + x^{7} + x^{6} + x^{4} + x^{3} + x^{2} + x + 1$	50000	
	$x^{80} + x^{13} + x^{8} + x^{6} + x^{5} + x^{3} + x^{2} + x + 1$ $x^{80} + x^{13} + x^{10} + x^{9} + x^{8} + x^{5} + x^{3} + x^{2} + 1$	40000 50000	
	$x^{80} + x^{11} + x^{9} + x^{8} + x^{5} + x + 1$ $x^{80} + x^{11} + x^{9} + x^{8} + x^{6} + x^{5} + x^{4} + x + 1$ $x^{80} + x^{12} + x^{9} + x^{5} + x^{4} + x^{3} + 1$ $x^{80} + x^{12} + x^{10} + x^{9} + x^{7} + x^{6} + 1$ $x^{80} + x^{12} + x^{11} + x^{9} + x^{8} + x^{7} + x^{5} + x + 1$	30000 40000 50000	
	$x^{80} + x^{11} + x^{10} + x^{9} + x^{7} + x^{6} + x^{5} + x^{4} + x^{3} + x + 1$ $x^{80} + x^{11} + x^{10} + x^{9} + x^{8} + x^{7} + x^{6} + x^{4} + x^{2} + x + 1$ $x^{80} + x^{12} + x^{9} + x^{6} + x^{5} + x + 1$ $x^{80} + x^{12} + x^{10} + x^{9} + x^{7} + x^{6} + x^{4} + x^{2} + 1$ $x^{80} + x^{13} + x^{10} + x^{7} + x^{5} + x^{3} + x^{2} + x + 1$ $x^{80} + x^{13} + x^{10} + x^{9} + x^{8} + x^{5} + x^{4} + x + 1$	20000 30000 40000 50000	

[표 4. (80차) NS]

비트 (차)	원시다항식	수열	초기치
96	$x^{96} + x^7 + x^6 + x^4 + x^3 + x^2 + 1$ $x^{96} + x^{11} + x^9 + x^7 + x^3 + x^2 + 1$ $x^{96} + x^{12} + x^{11} + x^8 + x^5 + x^3 + x^2 + x + 1$ $x^{96} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^3 + x^2 + 1$ $x^{96} + x^{13} + x^{10} + x^9 + x^8 + x^7 + x^4 + x^2 + 1$ $x^{96} + x^9 + x^7 + x^5 + x^2 + x + 1$ $x^{96} + x^{11} + x^{10} + x^5 + 1$ $x^{96} + x^{12} + x^{10} + x^8 + x^7 + x^6 + x^5 + x^4 + 1$ $x^{96} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^4 + x^3 + x + 1$	100000	3
	$x^{96} + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ $x^{96} + x^{10} + x^9 + x^7 + x^6 + x^3 + 1$ $x^{96} + x^{11} + x^{10} + x^6 + x^4 + x^3 + 1$ $x^{96} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^3 + x^2 + 1$ $x^{96} + x^{12} + x^{11} + x^6 + x^5 + x^3 + 1$ $x^{96} + x^{12} + x^{11} + x^{10} + x^8 + x^5 + x^2 + x + 1$ $x^{96} + x^{12} + x^{11} + x^{10} + x^9 + x^5 + x^2 + x + 1$ $x^{96} + x^{13} + x^9 + x^7 + x^6 + x^5 + x^2 + x + 1$ $x^{96} + x^{13} + x^{10} + x^9 + x^7 + x^6 + x^5 + x^3 + x^2 + x + 1$ $x^{96} + x^{13} + x^{10} + x^9 + x^8 + x^3 + x^2 + x + 1$	90000 100000	
	$x^{96} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^3 + x + 1$ $x^{96} + x^{12} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^4 + x^3 + x^2 + 1$ $x^{96} + x^{13} + x^{10} + x^8 + x^7 + x^6 + x^5 + x^3 + 1$	50000 80000 90000 100000	

[표 5. (96차) NS 1]

비트 (차)	원시다항식	수열	초기치
96	$x^{96} + x^{12} + x^{11} + x^{10} + x^{4} + x^{3} + x^{2} + x + 1$ $x^{96} + x^{13} + x^{10} + x^{6} + x^{5} + x^{4} + x^{3} + x + 1$ $x^{96} + x^{13} + x^{10} + x^{8} + x^{7} + x^{5} + x^{2} + x + 1$	30000 50000 80000 90000 100000	3
	$x^{96} + x^{13} + x^{10} + x^{7} + x^{6} + x + 1$	20000 30000 50000 80000 90000 100000	
	$x^{96} + x^{12} + x^{11} + x^{7} + x^{6} + x^{5} + 1$	19000 90000 100000 20000 30000 50000 80000	

[표 6. (96차) NS 2]

비트 (차)	원시다항식	수열	초기치
112	$x^{112} + x^9 + x^7 + x^4 + x^2 + x + 1$	30000	3
	$x^{112} + x^8 + x^6 + x^3 + x^2 + x + 1$ $x^{112} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + x + 1$	60000	
	$x^{112} + x^{11} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$	50000	
	$x^{112} + x^{10} + x^9 + x^7 + x^6 + x^4 + 1$	100000	
	$x^{128} + x^7 + x^2 + x + 1$	40000	
	$x^{128} + x^{10} + x^9 + x^7 + x^6 + x^5 + x^4 + x + 1$	100000	
128	$x^{128} + x^8 + x^6 + x^5 + x^4 + x + 1$	140000	71, 87, 99, 115
	$x^{128} + x^8 + x^6 + x^5 + x^4 + x^2 + 1$ $x^{128} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$	150000	3
192	$x^{192} + x^9 + x^8 + x^6 + x^5 + x + 1$	180000	3

[표 7. (112, 128, 192차) NS]

나. 결과의 활용

- 이번 LFSR(Linear Feedback Shift Register) 을 이용한 의사난수 구현 프로그램 과 통계 프로그램시스템들은 의사난수의 적합성을 평가하는데 활용하는 한편 다른 의사난수 생성기법 등을 추가 개발하는 기본 시스템으로 활용할 수 있다. 또한 의사난수 생성기의 보안성을 높이는 기술연구의 기본 자료로도 활용 할 수 있다.

4. 참고문헌

(1) H. S. Kim, J. K. Lee, and S. Kang, "A New Multiple Weight Set Calculation Algorithm," Proc. of International Test Conference, pp. 878-884, 2001

(2) http://blog.naver.com/f405?Redirect=Log&logNo=100010731162

(3) http://ko.wikipedia.org/wiki/%EB%A9%94%EB%A5%B4%EC%84%BC_%ED%8A%B8%EC %9C%84%EC%8A%A4%ED%84%B0

(4) http://ko.wikipedia.org/wiki/%EC%84%A0%ED%98%95_%ED%95%A9%EB %8F%99_%EC %83%9D%EC%84%B1%EA%B8%B0

(5) http://ko.wikipedia.org/wiki/%EC%97%AD_%ED%95%A9%EB%8F%99_%EC%83%9D%EC %84%B1%EA%B8%B0

(6) http://ko.wikipedia.org/wiki/RANDU

(7) http://ko.wikipedia.org/wiki/RANDU

(8) http://mwultong.blogspot.com/2008/03/pseudo-random-uniform-random-number.html

(9) http://seanerikoconnor.freeservers.com/Mathematics/AbstractAlgebra/PrimitivePolynomials/ Project/SourceCode/

(10) J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Cryptanalytic Attacks on Pseudo random Number Generators, In Proceedings of the Fast software Encryption", Fifth International Workshop, Springer-Verlag, Mar 1998.

(11) M. Matsumoto & T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator", ACM Trans.

Model. Comput. Simul. 8, 3 (1998).
(12) LavaRnd Random Number generator, http://www.lavarand.com
(13) Random.org generator http://www.random.org
(14) R. Lidl and H. Niederreiter, Introduction to Finite Fields and Their Application, Cam-bridge University Press. Cambridge, 1986
(15) W. Meier and O. Staffelbach. "Fast Cor-relation Attacks on Certain Stream Cipher". J. Cryptology (1). pp.159~176, 1989.
(16) William Stalling, "Cryptography and Network Security : Principles and Practices. Third Edtion", Prentice Hall, 2003.
(17) 김홍식, "3값 가중치 기법과 가변 순위 LFSR을 이용한 테스트 압축 기법", "연세대학교 대학원", p6~5, 2004.
(18) 박광현, "스트림암호 시스템", 133,134p, 2010.1.
(19) 박성준, "표준전자서명용 의사난수 생성 알고리즘개발", "한국 정보보호 진흥원", p4,p13,p15, 1999. 12
(20) 백경희, "LFSR기반 스트림암호에 관한 이해", "아주대 정보통신대학원", p3~4, p9, 2007. 2
(21) 안성수, 임광철, "암호알고리즘을 이용한 의사난수생성기의 구현", p425~428, 2003
(22) 양정모, "암호학", 경문사 144~158p ,183~185p 2010.8.30.
(23) 이정희, "타임스탬프 카운터 레지스터를 사용한 난수 발생기", "홍익대학교 대학원", p1~p3, 2004.
(24) 인준환, "카오스 어트렉터를 이용한 암호 알고리즘 연구", "명지대학교 대학원", p8,p29~33, 2008.
(25) 유혜정 , "차세대 암호알고리즘의 기반논리 통합검증도구 개발", "한국 정보보호 진흥원", p23,p19,p82~83, 2004. 12
(26) 최희봉, 원동호, "$GF(q)$상의 원시다항식 생성에 관한 연구", 2001. 2

5. 부록 : 소스

가. LFSR 생성기

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

FILE  *ftap, *frnd;

int main(void)
{
        unsigned short SR[513]={0},TAP[513];
        unsigned short feedback;
        char  buffer[2000], tmp[]="              ";
        char  ppfile[14], rndfile[15];
        char  *ptr_buffer;
        int   i, j, k;
    int   xx, yy, zz;
    int   eof, nbit, current_pp;
        int   addr_pp[1000];
        int   period, total_rnd, sr_count;


p1:                                                          // LFSR 비트 수 입력
        printf("\n>> LFSR의 비트 수를 입력하세요 (4 - 512) : ");
        scanf("%d", &nbit);
        if (nbit>=4 || nbit<=512) goto p2;
        printf(".. 입력 오류입니다. 재입력하세요.\n"); goto p1;

p2:
        xx=nbit/100; yy=(nbit%100)/10; zz=nbit%10;
        sprintf(ppfile,"ppfile%1d%1d%1d.txt",xx, yy, zz);
        sprintf(rndfile,"rndfile%1d%1d%1d.txt",xx, yy, zz);

        printf("\n.. 원시 다항식의 파일명칭은 %s 입니다.\n", ppfile);
        if ((ftap=fopen(ppfile,"r"))==NULL)
                {
                        printf(".. 파일 오픈 오류가 발생했습니다!\n");
                        goto end;
```

```
                }


        eof=0;
        for (i=1; i<=nbit; i++) TAP[i]=0;          // 탭 초기화
loop:
        printf("\n.. 원시 다항식은 아래와 같습니다.\n");

        for (i=1; i<=100; i++)
    {
                addr_pp[i]=ftell(ftap);
                if(fscanf(ftap, "%[^\n]", buffer)!=-1) goto skip;
                eof=1; if (i==1) eof=2;
                goto endoffile;
skip:
                ptr_buffer=&buffer[0];
                j=strcspn(ptr_buffer, ">"); k=strcspn(ptr_buffer, ",");
                strncpy((char *)ptr_buffer, (char *)(ptr_buffer+j+2), k-j-2);
                memset(ptr_buffer+k-j-2, '\0', 1);
                printf("%d : %s\n", i, ptr_buffer);
                fseek(ftap, ftell(ftap)+2,0);
        }

endoffile:                                          // 원시다항식 선택 및 탭 세팅
        if (eof==2) { printf("\n>> 작업할 원시다항식이 없습니다.\n"); goto end; }

ag:     printf("\n>> 원시 다항식을 선택하십시오 ( 선택 : 해당번호, 통과 : 0 ) ");
        scanf("%d", &current_pp);
    if (current_pp==0) goto next;
        if (current_pp<1 || current_pp>=i)
                {
                        printf(".. 선택 오류입니다.\n");
                        goto ag;
                }
        fseek(ftap, addr_pp[current_pp],0);
        fscanf(ftap, "%[^\n]", buffer);
        ptr_buffer=&buffer[0];
        j=strcspn(ptr_buffer, ">");
        k=strcspn(ptr_buffer, ",");
        strncpy((char *)ptr_buffer, (char *)(ptr_buffer+j+2), k-j-2);
        memset(ptr_buffer+k-j-2, '\0', 1);
        printf("\n.. 선택한 원시 다항식은 %s입니다.\n", ptr_buffer);
        fclose(ftap);
```

```
        while (1)
        {
                i=strcspn(ptr_buffer, "x");
                j=strcspn(ptr_buffer, "^");
                k=strcspn(ptr_buffer, "+");
                if ((i==0) && (j>=k))
                        {
                                xx=1;
                                goto x1;
                        }
                if ((i==j) && (j==k)) break;
                strncpy(tmp, (char *)(ptr_buffer+j+1),(k-j-1));

                xx=atoi(tmp);
x1:
                strcpy((char *)ptr_buffer, (char *)(ptr_buffer+k+2));
                TAP[xx]=1;
        }
        printf(".. 원시 다항식에 의거 설정된 탭은 아래와 같습니다.\n");
        for (i=1; i<=nbit; i++)
        {
                printf("%d", TAP[i]);
                if ((((nbit-i)%4)==0)&&(i!=nbit)) printf("_");
        }
        printf("\n");
        goto p3;

next:
        if (eof==0) goto loop;
        if (eof==1)
                {
                        printf(".. 추가 작업할 원시다항식이 없습니다.\n");
                        goto end;
                }
        printf(".. 정상 작업상태가 아닙니다.\n"); goto end;

p3:
        period=-1;
        if(nbit<32) period=(1<<nbit)-1;
        printf(".. LFSR의 주기는 2^%d - 1입니다.\n", nbit);

        printf("\n>> 생성할 난수의 갯수를 입력하세요 ? ");
        scanf("%d", &total_rnd);
```

```
        printf("\n>> LFSR의 초기치를 입력하세요 ");
        scanf("%d", &xx);

        i=0;
while(1){
                SR[i]=((xx>>i)&1);
                printf("\n%d",SR[i]);
                        i++;
                        if(xx>>i==0)break;
        }

if ((frnd=fopen(rndfile,"w"))==NULL)
                {
                        printf(".. 파일 오픈 오류가 발생했습니다!\n");
                        goto end;
                }


        fprintf(frnd, "%3d %10d %10d\n", nbit, total_rnd, period);
        printf(".. LFSR 비트 수는 %d이고,  생성할 수열의 길이는 %d입니다.\n", nbit, total_rnd);

        if ((nbit<32) && (total_rnd>period)) total_rnd=period;

        printf(".. 설정한 탭은 아래와 같습니다.\n");
        for (i=1; i<=nbit; i++)
        {
                fprintf(frnd, "%1d", TAP[i]);
                printf("%1d", TAP[i]);
        }
        fprintf(frnd, "\n"); printf("\n");
        printf(".. LFSR의 초기치는 아래와 같습니다.\n");
        for (i=0; i<=nbit; i++)
        {
                fprintf(frnd, "%1d", SR[i]);
                printf("%1d", SR[i]);
        }
        fprintf(frnd, "\n"); printf("\n");

/*      run NBIT bits LFSR */

        printf(".. 생성한 LFSR의 수열은 아래와 같습니다.\n");
        sr_count=0;

        while (1)
```

```
	{
		feedback=0;
		for(j=nbit; j>=1; j--)
		{
			if(TAP[j]==1) feedback=feedback ^ SR[j];
		}
		fprintf(frnd, "%1d", SR[nbit]);

		if (((++sr_count)%60)==0)
		{
			fprintf(frnd,"\n");
			printf("  생성 비트 %6d\n", sr_count);
		}

		for (j=nbit; j>1; j--) SR[j]=SR[j-1];		//  shift right 1 bit of
SR
		SR[1]=feedback;
		if(sr_count>=total_rnd)
		{
			printf("\n.. 생성한 LFSR의 수열은 총 %d 비트입니다.", sr_count);
			break;
		}
	}
	fclose(frnd);

end:
	printf("\n\n>> LFSR에 의한 난수 생성작업이 모두 끝났습니다!!\n");
}
```

나. 도수-m 검정(중첩, 도수검정 포함)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

int     polygram[9][256], buff[9], idx[9], idv[9];
int     mask[9]={0,0,1,3,7,15,31,63,127};
double mean, sum, tsum;
double chisquare[9]={0.0,3.8414,7.8147,14.0671,24.9957,44.9853,82.5287,154.3015,292.1748};
FILE    *ftap, *frnd;

void calc_polygram(int onebyte)
{
        int i;

        for (i=1; i<=8; i++)
        {
                idv[i]=idv[i]*2+onebyte;
                idx[i]++;
                if (idx[i]==i)
                {
                        polygram[i][idv[i]]++;
                        idx[i]--;
                        idv[i]=idv[i]&mask[i];
                }
        }
}

void convert_binary(int poly_no, int page)
{
        int i, tmp;

        tmp=page;
        for (i=poly_no; i>=1; i--)
        {
                buff[i]=0;
                if ((tmp&1)==1) buff[i]=1;
                tmp=tmp>>1;
        }
}

void blankprint(int poly_no, int column)
{
        int i, blk;
```

```
        blk=(80-column*18)/2+(column-4)*1+4;
        printf("\n");
        for (i=1; i<blk; i++) printf(" ");
}

int main(void)
{
        unsigned short SR[513], TAP[513];
        char   tmp[]="              ", rndfile[15];
        int    i, j, k, l, xx, yy, zz, nbit, onebyte, period, skip, total_rnd;

        for (i=1; i<=8; i++)
        {
                idx[i]=0;
                idv[i]=0;
                for (j=0; j<256;j++) polygram[i][j]=0;
        }

p1:
        printf("\n>> 검정대상 LFSR의 비트 수를 입력하세요 (4 - 512) : ");
        scanf("%d", &nbit);
        if (nbit>=4 || nbit<=512) goto p2;
        printf(".. 입력 오류입니다. 재입력하세요.\n"); goto p1;

p2:
        xx=nbit/100; yy=(nbit%100)/10; zz=nbit%10;
        sprintf(rndfile,"rndfile%1d%1d%1d.txt",xx, yy, zz);

        printf("\n.. 검정대상 파일명칭은 %s 입니다.\n", rndfile);
        if ((frnd=fopen(rndfile,"r"))==NULL)
                {
                        printf(".. 파일 오픈 오류가 발생했습니다!\n");
                        goto end;
                }
        fscanf(frnd, "%3d %10d %10d", &nbit, &total_rnd, &period);

        for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &TAP[i]);
        for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &SR[i]);

        if ((total_rnd>period) && (period!=-1))
        {
                total_rnd=period;
                printf("\n.. 난수 주기가 짧아 주기와 동일한 길이의 난수열만 검정합니다.\n");
        }
```

```
for (i=1; i<=total_rnd; i++)
{
        fscanf(frnd, "%1d", &onebyte);
        calc_polygram(onebyte);
}
fclose(frnd);

printf("\n");
for (i=1,zz=0; i<=8; i++)
{
        onebyte=1<<i;
        skip=2;
        if (onebyte>4) skip=4;
        if (i==1)
        {
                printf("\n\n\t\t\t  도수 ( m - %d ) 검정 결과", i);
                printf("\n\t\t\t  ========================\n");
        }
        else
        {
                printf("\n\n\t\t     도수 ( m - %d ) 검정 결과(블록 중첩식)", i);
                printf("\n\t\t     ======================================\n");
        }
        if (i<3)
        {
                blankprint(i, skip); for(j=1;j<=31;j++)printf("-");
                blankprint(i, skip); printf("2진법    빈도          2진법    빈도");
                blankprint(i, skip); for(j=1;j<=31;j++)printf("-");
        }
        else
        {
                blankprint(i, skip);
                for(j=1;j<=65;j++)printf("-");
                blankprint(i, skip);
                printf(" 2진법     빈도     2진법     빈도     2진법     빈도     2진
법     빈도");
                blankprint(i, skip);
                for(j=1;j<=65;j++)printf("-");
        }

        for (j=0; j<(onebyte/skip); j++)
        {
                blankprint(i, skip);
                if (skip==2) printf(" ");
                for (k=0; k<skip; k++)
                {
```

```
                                        convert_binary(i, j+(onebyte/skip)*k);
                                        if (k!=0) printf("    ");
                                        for (l=1; l<=i; l++) printf("%1d", buff[l]);
                                        if (skip==2) for (xx=1; xx<6-i; xx++) printf(" ");
                                        if (skip==4) for (xx=1; xx<=(8-i); xx++) printf(" ");
                                        printf(" %5d", polygram[i][j+(onebyte/skip)*k]);
                                        if (skip==2) printf("       ");
                                }
                        }
                        blankprint(i, skip);
                        if (i<3)for(j=1;j<=31;j++)printf("-");
                        else for(j=1;j<=65;j++)printf("-");
                        printf("\n");

                        sum=0.0;
                        mean=(double)(total_rnd-i+1)/onebyte;
                        for (j=0; j<onebyte; j++)
                        {
                                tsum=(double)polygram[i][j]-(double)mean;
                                sum=sum+tsum*tsum;
                        }
                        sum=sum/mean;
                        printf("\n\t\t    생성된 난수의 중첩 %d-도수별 기대 값  = %7.3f\n",i ,mean);
                        printf("\n\t\t    생성된 난수의 카이제곱 값      = %7.3f\n", sum);
                        printf("\n\t\t      자유도 %3d의 카이제곱 기준 값 = %7.3f\n", onebyte-1,
chisquare[i]);
                        if(sum<chisquare[i])++zz;
                }
                printf("\n\t\t%d비트 에서 중첩m-도수 검정을 총 [%0.0f%%]통과하였습니다.\n",nbit,((double)zz/8.0)*100);
                printf("\n.. 설정된 탭은 아래와 같습니다.\n");
                for (i=1; i<=nbit; i++) printf("%1d", TAP[i]);
                printf("\n.. LFSR의 초기치는 아래와 같습니다.\n");
                for (i=1; i<=nbit; i++) printf("%1d", SR[i]);
                printf("\n.. 검정용 난수는 총 %d 비트입니다.\n", total_rnd);
end:
                printf("\n\n>> LFSR에 의해 생성된 난수의 중첩식 도수 - m 검정 작업이 끝났습니다!!\n");
}
```

다. 도수-m 검정(비중첩, 도수검정 포함)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

int     polygram[9][256], buff[9], idx[9], idv[9], idn[9];
double mean, sum, tsum;
double chisquare[9]={0.0,3.8414,7.8147,14.0671,24.9957,44.9853,82.5287,154.3015,292.1748};
FILE    *ftap, *frnd;

//비중첩 수열 읽어오기
void calc_polygram_nonover(int onebyte)
{
        int i;

        for (i=1; i<=8; i++)
        {
                idv[i]=idv[i]*2+onebyte;
                idx[i]++;
                if (idx[i]==i)
                {
                        polygram[i][idv[i]]++;
                        idx[i]=0;
                        idv[i]=0;
                        idn[i]++;
                }
        }
}

void convert_binary(int poly_no, int page)
{
        int i, tmp;

        tmp=page;
        for (i=poly_no; i>=1; i--)
        {
                buff[i]=0;
                if ((tmp&1)==1) buff[i]=1;
                tmp=tmp>>1;
        }
}
```

```
void blankprint(int poly_no, int column)
{
        int i, blk;

        blk=(80-column*18)/2+(column-4)*1+4;
        printf("\n");
        for (i=1; i<blk; i++) printf(" ");
}

int main(void)
{
        unsigned short SR[513], TAP[513];
        char   tmp[]="              ", rndfile[15];
        int    i, j, k, l, xx, yy, zz, nbit, onebyte, period, skip, total_rnd;

        for (i=1; i<=8; i++)
        {
                idx[i]=0;
                idv[i]=0;
                idn[i]=0;
                for (j=0; j<256;j++) polygram[i][j]=0;
        }

p1:
        printf("\n>> 검정대상 LFSR의 비트 수를 입력하세요 (4 - 512) : ");
        scanf("%d", &nbit);
        if (nbit>=4 || nbit<=512) goto p2;
        printf(".. 입력 오류입니다. 재입력하세요.\n"); goto p1;

p2:
        xx=nbit/100; yy=(nbit%100)/10; zz=nbit%10;
        sprintf(rndfile,"rndfile%1d%1d%1d.txt",xx, yy, zz);

        printf("\n.. 검정대상 파일명칭은 %s 입니다.\n", rndfile);
        if ((frnd=fopen(rndfile,"r"))==NULL)
                {
                        printf(".. 파일 오픈 오류가 발생했습니다!\n");
                        goto end;
                }
        fscanf(frnd, "%3d %10d %10d", &nbit, &total_rnd, &period);

        for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &TAP[i]);
        for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &SR[i]);
```

```
if ((total_rnd>period) && (period!=-1))
{
        total_rnd=period;
        printf("\n.. 난수 주기가 짧아 주기와 동일한 길이의 난수열만 검정합니다.\n");
}

for (i=1; i<=total_rnd; i++)
{
        fscanf(frnd, "%1d", &onebyte);
        calc_polygram_nonover(onebyte);
}
fclose(frnd);

printf("\n");
for (i=1,zz=0; i<=8; i++)
{
        onebyte=1<<i;
        skip=2;
        if (onebyte>4) skip=4;

        if (i==1)
        {
                printf("\n\n\t\t\t  도수 ( m - %d ) 검정 결과", i);
                printf("\n\t\t\t  ========================\n");
        }
        else
        {
                printf("\n\n\t\t   도수 ( m - %d ) 검정 결과(블록 비중첩식)", i);
                printf("\n\t\t   ========================================\n");
        }
        if (i<3)
        {
                blankprint(i, skip); for(j=1;j<=31;j++)printf("-");
                blankprint(i, skip); printf("2진법    빈도          2진법    빈도");
                blankprint(i, skip); for(j=1;j<=31;j++)printf("-");
        }
        else
        {
                blankprint(i, skip);
                for(j=1;j<=65;j++)printf("-");
                blankprint(i, skip);
                printf(" 2진법     빈도     2진법     빈도     2진법     빈도     2진
```

```
법      빈도");
                        blankprint(i, skip);
                        for(j=1;j<=65;j++)printf("-");

                }

                for (j=0; j<(onebyte/skip); j++)
                {
                        blankprint(i, skip);
                        if (skip==2) printf(" ");
                        for (k=0; k<skip; k++)
                        {
                                convert_binary(i, j+(onebyte/skip)*k);
                                if (k!=0) printf("    ");
                                for (l=1; l<=i; l++) printf("%1d", buff[l]);
                                if (skip==2) for (xx=1; xx<6-i; xx++) printf(" ");
                                if (skip==4) for (xx=1; xx<=(8-i); xx++) printf(" ");
                                printf(" %5d", polygram[i][j+(onebyte/skip)*k]);
                                if (skip==2) printf("       ");
                        }
                }
                blankprint(i, skip);
                if (i<3)for(j=1;j<=31;j++)printf("-");
                else for(j=1;j<=65;j++)printf("-");
                printf("\n");

                sum=0.0;
                mean=(double)idn[i]/onebyte;
                for (j=0; j<onebyte; j++)
                {
                        tsum=(double)polygram[i][j]-(double)mean;
                        sum=sum+tsum*tsum;
                }
                sum=sum/mean;
                printf("\n\t\t    생성된 난수의 비중첩 %d-도수별 도수별 기대 값  = %7.3f\n",
i,mean);
                printf("\n\t\t    생성된 난수의 카이제곱 값       = %7.3f\n", sum);
                printf("\n\t\t      자유도 %3d의 카이제곱 기준 값 = %7.3f\n", onebyte-1,
chisquare[i]);
                if(sum<chisquare[i])++zz;
        }
        printf("\n\t\t%d비트     에서     비중첩m-도수     검정을     총     [%0.0f%%]통과하였습니
다.\n",nbit,((double)zz/8.0)*100);
        printf("\n.. 설정된 탭은 아래와 같습니다.\n");
```

```
        for (i=1; i<=nbit; i++) printf("%1d", TAP[i]);
        printf("\n.. LFSR의 초기치는 아래와 같습니다.\n");
        for (i=1; i<=nbit; i++) printf("%1d", SR[i]);
        printf("\n.. 검정용 난수는 총 %d 비트입니다.\n", total_rnd);
end:
        printf("\n\n>> LFSR에 의해 생성된 난수의 비중첩식 도수 - m 검정 작업이 끝났습니다!!\n");
}
```

라. 포커검정

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

int     polygram[9][256]={0}, idx[9]={0}, idv[9]={0},idn[256]={0};
int     from=1, to=8, no_one;
double  freq[9][257]={0},chisqure[9]={0,3.8414,5.9914,7.8147,9.4877,
                                        11.0704,12.5915,14.0671,15.5073};
FILE    *ftap, *frnd;

void calc_polygram_nonover(int onebyte)
{
        int i;

        for (i=from; i<=to; i++)
        {
                idv[i]=idv[i]+onebyte;
                idx[i]++;
                if (idx[i]==i)
                {
                        polygram[i][idv[i]]++;
                        idx[i]=0;
                        idv[i]=0;
                        idn[i]++;
                }
        }
}

void blankprint(int poly_no, int column)
{
        int i, blk;

        blk=(80-column*18)/2+(column-4)*1+4;
        printf("\n");
        for (i=1; i<blk; i++) printf(" "); // 들여쓰기 1~2=24 , 3~8= 8
}

int main(void)
{
        unsigned short SR[513], TAP[513];
```

```
        char   tmp[]="              ", rndfile[15];
        int    i, j, k, l, xx, yy, zz, count,bit;
    int    nbit, onebyte, skip=4, total_rnd;
        double sum, range;

p1:                                                                    // LFSR 비트 수 입력
        printf("\n>> 검정대상 LFSR의 비트 수를 입력하세요 (4 - 512) : "); // 입력구간
        scanf("%d", &nbit);
        if (nbit>=4 || nbit<=512) goto p2;
        printf(".. 입력 오류입니다. 재입력하세요.\n"); goto p1;

p2:
        xx=nbit/100; yy=(nbit%100)/10; zz=nbit%10;
        sprintf(rndfile,"rndfile%1d%1d%1d.txt",xx, yy, zz);
        printf("\n.. 검정대상 파일명칭은 %s 입니다.\n", rndfile);
        if ((frnd=fopen(rndfile,"r"))==NULL)
                {
                        printf(".. 파일 오픈 오류가 발생했습니다!\n");
                        goto end;
                }
        fscanf(frnd, "%3d %10d %10d", &nbit, &total_rnd, &zz);

        for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &TAP[i]);
        for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &SR[i]);


        for (i=1; i<=total_rnd; i++)
        {
                fscanf(frnd,"%1d",&onebyte);  // 난수 한 비트 값 읽기
                calc_polygram_nonover(onebyte);     // 도수-from부터 도수-to까지 빈도 계산
                }
        fclose(frnd);
        printf("\n");

        for (i=from,xx=0; i<=to; i++)
        {
                onebyte=1<<i;

                printf("\n");
                for(k=1;k<=30;k++)printf(" ");
                printf("Pocker - %d 검정 결과\n",i);
                for(k=1;k<=30;k++)printf(" ");
                for(k=1;k<=20;k++)printf("=");
```

```
if(i<=3){
        blankprint(i, skip);
        printf("\t\t\t          1의 갯수      빈도");
        blankprint(i, skip);
        printf("\t\t\t    -------------------------");
}
if(i>3){
        blankprint(i, skip);
        printf("   1의 갯수     빈도\t 1의 갯수     빈도   \t1의 갯수     빈도
");
        blankprint(i, skip);
        if(i>3)for(j=1;j<=67;j++)printf("-");
}
printf("\n");

no_one=i+1;                    range=(int)pow(2.0,i); // * 기대치 분석 *


for (k=0; k<range; k++){

        for (l=0,bit=k,count=0; l<i; l++, bit=bit>>1){

                if ((bit&0x01)==1) count++;

        }
        freq[i][count]++;
}
 for (j=0; j<no_one; j++){
         k=(int)freq[i][j];
         freq[i][j]=((float)k/range)*(float)idn[i]; // (n(i)/2^m) * (total/i)

 }
 for (j=0; j<=i; j++){

         if(i<=3){
                 printf("\t\t\t%11d\t%7d",j,polygram[i][j]);
                 if(j==i)continue;printf("\n");
         }
         if(i>3){
                 printf("\t%5d\t%7d   ",j,polygram[i][j]);


                 if(i==j)continue;if(j==2||j==5)printf("\n");
         }
```

```
                    }

                    for (j=0,sum=0.0; j<=i; j++){    // *시그마 계산*
                              sum=(pow(polygram[i][j]-freq[i][j],2)/freq[i][j])+sum;
                    }
                    blankprint(i, skip);
                    if(i<=3){
                              printf("\t\t\t     ");for(j=1;j<=25;j++)printf("-");
                             }
                    if(i>3)for(j=1;j<=67;j++)printf("-");
                    printf("\n                              생성된 Poker 검정의 카이제곱 값        =
%7.3f\n", sum);
                    printf("\n                         자유도 %3d의 카이제곱 기준 값 = %7.3f\n\n",
onebyte-1, chisqure[i]);
                    if(sum<chisqure[i])++xx;
          }
          printf("\t\t%d비트        에서        Pocker        검정을        총        [%0.0f%%]통과하였습니
다.\n",nbit,((double)xx/8.0)*100);

          blankprint(i, skip);
          printf("\n.. 설정된 탭은 아래와 같습니다.\n");
          for (i=1; i<=nbit; i++)printf("%1d", TAP[i]);
          printf("\n.. LFSR의 초기치는 아래와 같습니다.\n");
          for (i=1; i<=nbit; i++)printf("%1d", SR[i]);
          printf("\n.. 검정용 난수는 총 %d 비트입니다.\n", total_rnd);
end:
          printf("\n\n>> LFSR에 의해 생성된 난수 검정 작업이 끝났습니다!!\n");
}
```

마. 런 검정(길이검정, 수 검정)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
int      B[9][256]={0}, G[9][256]={0}, change[9][9]={0};
int      seq, state_ln=-1, state_no=-1, run_no;
int      onebyte, p_check=0, p_count=0, count;
double sum=0.0;
double chisquare[]={0.0,3.84,5.99,7.81,9.49,11.07,12.59,14.07,15.51,16.92,18.31,19.68,
            21.03,22.36,23.68,25,26.30,27.59,28.87,30.14,31.41,32.67,33.92,
            35.17,36.42,37.65,38.89,40.11,41.34,42.56,43.77,55.76,67.50,79.08,
            90.53,101.88,113.14,124.34};
float   run_len_exp[9],num_exp[9];;
FILE   *ftap, *frnd;
void calculate_run_number_exp(int i,int total_rnd) //런의 수 검정 기대값
{
 int k,j,run;
 for(k=0;k<=9;k++)num_exp[k]=0; sum=(int)pow(2.0,i);
 for (k=0; k<sum; k++){
   for (j=0,onebyte=k,run=0,seq=-1; j<i; j++, onebyte=onebyte>>1){
    if ((onebyte&0x01)==1){ if(seq==1)continue;run++; seq=1; }
    if ((onebyte&0x01)==0){ if(seq==0)continue;run++; seq=0; }
   }
   num_exp[run]++;
 }
 for (j=1; j<i+1; j++){
   k=(int)num_exp[j];
   num_exp[j]=((float)k/sum)*((float)total_rnd/i);
 }
 sum=0.0;
}

void calculate_run_exp(int run_length)
{
 int     i;
   for (i=1; i<=run_length; i++)
 {
   sum=(float)pow(2., i+1);
   if (i==run_length) { run_len_exp[i]=(float)2./sum; break; }
   run_len_exp[i]=((((float)run_length-i-(float)1.)/(float)2.)+(float)2.)/sum;
 }
```

```
 sum=0.0;
}
void RUN(int index,int total_rnd)
{
 int l,p;
 seq=0; state_ln=-1; state_no=-1; run_no=0;
 for(l=1; l<=(total_rnd/index); l++)
 {
  while(1)
  {
   for(p=1; p<=index; p++)
   {
    fscanf(frnd,"%1d",&onebyte);
    if(p_count<0) goto print_skip;
print_skip:   switch(onebyte)
    {
 case 0:    if(state_ln==1)
      {
       B[index][seq]++; seq=0;
      }
      seq++; state_ln=0;
      if(state_no==1) run_no++;
      state_no=0;
      break;
 case 1:    if(state_ln==0)
      {
       G[index][seq]++; seq=0;
      }
      seq++; state_ln=1;
      if(state_no==0) run_no++;
      state_no=1;
      break;
    }
   }
   change[index][run_no+1]++;
   state_no=-1; run_no=0;
   if(state_ln==0) G[index][seq]++;
   if(state_ln==1) B[index][seq]++;
   seq=0; state_ln=-1;
   break;
  }
 }
 p_count=-1;
}
```

```
void main(void)
{
 unsigned short SR[513], TAP[513];
 char   tmp[]="              ", rndfile[15], buff[1000];
 int    i, xx, yy, zz, nbit,   period, total_rnd;
 long double EA;
p1:
 printf("\n>> 검정대상 LFSR의 비트 수를 입력하세요 (4 - 512) : ");
 scanf("%d", &nbit);
 if (nbit>=4 || nbit<=512) goto p2;
 printf(".. 입력 오류입니다. 재입력하세요.\n"); goto p1;
p2:
 xx=nbit/100; yy=(nbit%100)/10; zz=nbit%10;
 sprintf(rndfile,"rndfile%1d%1d%1d.txt",xx, yy, zz);
 printf("\n.. 검정대상 파일명칭은 %s 입니다.\n", rndfile);
 if ((frnd=fopen(rndfile,"r"))==NULL)
  {
   printf(".. 파일 오픈 오류가 발생했습니다!\n");
   goto end;
  }
 fscanf(frnd, "%3d %10d %10d", &nbit, &total_rnd, &period);
 for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &TAP[i]);
 for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &SR[i]);
 if ((total_rnd>period) && (period!=-1))
 {
  total_rnd=period;
  printf("\n.. 난수 주기가 짧아 주기와 동일한 길이의 난수열만 검정합니다.\n");
 }
 for(i=2; i<=8; i++)
 {
  fseek(frnd,0,0);
  fgets(buff,sizeof(buff),frnd); fgets(buff,sizeof(buff),frnd);
  fgets(buff,sizeof(buff),frnd);
  RUN(i,total_rnd);
 }
 fclose(frnd);

 for(count=2,xx=0; count<=8; count++)
 {
  printf("\n\n                                 블록 - %d 런 검정 결과\n", count);
  printf("                                ======================\n");
  printf("\n     1. 런 길이 검정\n");
  p              r              i              n              t              f              (              "
-----------------------------------------------------------------------\n");
```

```
  printf("      Block 런 길이  측정 값 | Gap 런 길이   측정 값 | block/gap 기대 값\n");
  p             r             i             n             t             f             (             "
-----------------------------------------------------------------------\n");
  calculate_run_exp(count);

  for(i=1; i<=count; i++)
  {
   EA=((int)(total_rnd/count))*run_len_exp[i];
   sum=((pow((B[count][i]-EA),2))+(pow((G[count][i]-EA),2)))/EA+sum;
   printf("\t%4d\t %9d   | %7d\t%11d   |  %10.3f\n",i,B[count][i],i,G[count][i],EA);
  }
  p             r             i             n             t             f             (             "
-----------------------------------------------------------------------\n");
  printf("\n                    생성 난수의 런의 길이 카이제곱 값 = %7.3f\n\n\t\t   ", sum);
  printf("자유도 %3d의 카이제곱 기준 값   = %7.3f\n", (2*count)-2,chisquare[((2*count)-2)]);
  printf("\n     2. 런 수 검정 \n");
  p             r             i             n             t             f             (             "
-----------------------------------------------------------------------\n");
  printf("            런 수        |      런 수 측정 값      |       런 수 기대 값\n");
  p             r             i             n             t             f             (             "
-----------------------------------------------------------------------\n");
  if(sum<chisquare[((2*count)-2)])++xx;
  calculate_run_number_exp(count,total_rnd);

  for(i=1; i<=count; i++)
  {
   sum=pow(change[count][i]-num_exp[i],2)/num_exp[i]+sum;
   printf("\t%7d \t | \t%9d \t  | \t%10.3f\n", i, change[count][i], num_exp[i]);
  }
  p             r             i             n             t             f             (             "
-----------------------------------------------------------------------\n");
  printf("\n                    생성된 난수의 런의 수 카이제곱 값      = %7.3f\n", (float)sum);
  printf("\n\t\t   자유도 %3d의 카이제곱 기준 값 = %7.3f\n", count-1,chisquare[count-1]);
  if(sum<chisquare[count-1])++xx;
 }
 printf("\n\t\t%d비트 에서 런 검정을 총 [%0.0f%%]통과하였습니다.\n",nbit,((double)xx/14.0)*100);
 printf("\n.. 설정된 탭은 아래와 같습니다.\n");
 for (i=1; i<=nbit; i++) printf("%1d", TAP[i]);
 printf("\n.. LFSR의 초기치는 아래와 같습니다.\n");
 for (i=1; i<=nbit; i++) printf("%1d", SR[i]);
 printf("\n.. 검정용 난수는 총 %d 비트입니다.\n", total_rnd);
end:
 printf("\n\n>> LFSR에 의해 생성된 난수의 런 검정 작업이 끝났습니다!!\n");
}
```

바. 자기상관검정(계열검정 포함)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

int     gram[3][4], idx, idv, ptr_value[9];
int     d_value, currentptr, oldptr, idp, jdp;
int     n0, n1, n00, n01, n10, n11, record;
double serial, autocorrel;
double chisquare[9]={0.0,3.84,7.81,14.07,25.00,55.76,99.00,99.00,99.00};
FILE   *ftap, *frnd;
void calc_polygram(int onebyte, int d)
{
            if((currentptr-oldptr)==d)
            {
                        oldptr++;
                        idv=ptr_value[jdp]*2+onebyte;
                        jdp++; if (jdp>d) jdp=1;
                        idx++;
                        if (idx==2)
                        {
                                    gram[2][idv]++;
                                    idx--;
                        }
            }
            idp++; if (idp>d) idp=1;
            ptr_value[idp]=onebyte;
            currentptr++;
}

int main(void)
{
            unsigned short SR[513], TAP[513];
            char   tmp[]="              ", rndfile[15];
            int    i, j, xx, yy, zz;
     int    nbit, onebyte, period, total_rnd;
p1:
            printf("\n>> 검정대상 LFSR의 비트 수를 입력하세요 (4 - 512) : ");
            scanf("%d", &nbit);
            if (nbit>=4 || nbit<=512) goto p2;
            printf(".. 입력 오류입니다. 재입력하세요.\n"); goto p1;
```

```
p2:
        xx=nbit/100; yy=(nbit%100)/10; zz=nbit%10;
        sprintf(rndfile,"rndfile%1d%1d%1d.txt",xx, yy, zz);

        printf("\n.. 검정대상 파일명칭은 %s 입니다.\n", rndfile);
        if ((frnd=fopen(rndfile,"r"))==NULL)
                {
                        printf(".. 파일 오픈 오류가 발생했습니다!\n");
                        goto end;
                }
        fscanf(frnd, "%3d %10d %10d", &nbit, &total_rnd, &period);

        for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &TAP[i]);
        for (i=1; i<=nbit; i++) fscanf(frnd, "%1d", &SR[i]);

        if ((total_rnd>period) && (period!=-1))
        {
                total_rnd=period;
                printf("\n.. 난수 주기가 짧아 주기와 동일한 길이의 난수열만 검정합니다.\n");
        }

        record=ftell(frnd);
        for (d_value=1,xx=0; d_value<=8; d_value++)
        {
                fseek(frnd, record, 0);
                for (i=0; i<=2; i++) for (j=0; j<=3;j++) gram[i][j]=0;
                idx=1; idv=0; oldptr=1; currentptr=1; idp=9; jdp=1;

                for (i=1; i<=total_rnd; i++)
                {
                        fscanf(frnd, "%1d", &onebyte);

                        calc_polygram(onebyte, d_value);
                }
                printf("\n");
                n00=gram[2][0];    n01=gram[2][1];
                n10=gram[2][2];    n11=gram[2][3];
                n0=n00+n01;         n1=n10+n11;
                if (d_value!=1) goto autocorrelation;

//  계열 검정 처리 루틴
                printf("\n\n                                    계열(serial) 검정 결과");
                printf("\n                                  =======================\n\n");
printf(" ----------------------------------------------------------------------\n");
```

```
                    printf("        2진법     빈  도     |      2진법     빈  도         2진법
빈  도\n");
printf("-----------------------------------------------------------------------\n");
                    printf("         0 %10d      |       00 %10d          10 %10d\n", n0, n00,
n10);
                    printf("         1 %10d      |       01 %10d          11 %10d\n", n1, n01,
n11);
printf("-----------------------------------------------------------------------\n");


serial=4.*(pow((float)n00,2)+pow((float)n01,2)+pow((float)n10,2)+pow((float)n11,2))/(total_rnd-1.);
                    serial=serial-2.*(pow((float)n0,2)+pow((float)n1,2))/total_rnd+1.;

                    printf("\n\t\t    계열 검정의 카이제곱 값      = %7.3f\n", serial);
                    printf("\n\t\t    자유도 2의 카이제곱 기준 값 = %7.3f\n", chisquare[2]);
                    if(serial<chisquare[2])++xx;
                    continue;

//  자기상관 검정 루틴
autocorrelation:
                    printf("\n\n                                  자기상관( d = %d ) 검정 결과",
d_value);
                    printf("\n                             ==========================\n\n");
                    printf("
 -----------------------------------------------------------------------\n");
                    printf("        2진법     빈  도     |      2진법     빈  도         2진법
빈  도\n");
                    printf("
-----------------------------------------------------------------------\n");
                    printf("         0 %10d      |       00 %10d          10 %10d\n", n0, n00,
n10);
               printf("         1 %10d      |       01 %10d          11 %10d\n", n1, n01, n11);
                    printf("
-----------------------------------------------------------------------\n");


autocorrel=4.*(pow((float)n00,2)+pow((float)n01,2)+pow((float)n10,2)+pow((float)n11,2))/(total_rnd-1);
                    autocorrel=autocorrel-2.*(pow((float)n0,2)+pow((float)n1,2))/total_rnd+1.;

                    printf("\n\t\t    자기상관 검정의 카이제곱 값  = %7.3f\n", autocorrel);
                    printf("\n\t\t    자유도 2의 카이제곱 기준 값  = %7.3f\n", chisquare[2]);
                    if(autocorrel<chisquare[2])++xx;
          }
```

```
end:
        fclose(frnd);

        printf("\t\t%d비트 에서 자기상관&계열 검정을 총 [%0.0f%%]통과하였습니다.\n",nbit,((double)xx/8.0)*100);
                printf("\n.. 설정된 탭은 아래와 같습니다.\n");
                for (i=1; i<=nbit; i++) printf("%1d", TAP[i]);
                printf("\n.. LFSR의 초기치는 아래와 같습니다.\n");
                for (i=1; i<=nbit; i++) printf("%1d", SR[i]);
                printf("\n.. 검정용 난수는 총 %d 비트입니다.\n", total_rnd);
        printf("\n\n>> LFSR에 의해 생성된 난수의 계열 검정과 자기상관 검정 작업이 끝났습니다!!\n");
}
```

간지 넣어주세요

2010 졸업연구 결과보고서

안드로이드 모바일 O.T.P 어플리케이션 및 시스템 구현

팀 명	올드보이(OLDBOY)
지도교수	양 정 모 교수님
	김 범 한 교수님
팀 원	최 연 규(4학년)
	최 형 진(4학년)
	신 기 식(4학년)
	이 욱 진(4학년)
	김 선 일(4학년)
	박 하 나(4학년)
스 텝	신 용 준(3학년)

2010. 10

중부대학교 정보보호학과

요 약 문

1. 연구제목

안드로이드 모바일 O.T.P 어플리케이션 및 시스템 구현

2. 연구 목적 및 필요성

지식정보화 사회에서의 개인정보보안의 중요성을 알리고 그에 따라 최근 화두가 되고 있는 OTP 시스템 구현에 대하여 연구하였다.

OTP의 기본구조를 살피고 그중에서도 편의성을 가진 **안드로이드 모바일 O.T.P 어플리케이션**과 기본 작동 시스템을 만들어 실질적인 OTP의 발급과 사용까지 구현함으로써 앞으로 모든 분야에서의 성공적인 OTP 기술의 정착을 위해 고려되어야 할 요인들을 시사해보도록 한다.

3. 연구 내용

본 연구에서는 OTP의 기본 구조와 앞으로의 전망 등에 대한 이론적 부분과 OTP의 종류 중 편의성을 장점화한 모바일 OTP를 안드로이드 어플리케이션을 이용하여 구현함으로써 OTP의 전반적인 지식을 담고 있다.

4. 연구 결과

앞으로 OTP 시장의 규모는 엄청나게 커질 것이며 OTP가 가져올 기대효과 또한 엄청날 수 있다. OTP는 이제 생소하기 보다는 바로 실생활에서 흔히 사용하는 기술이 될 것이다. 하지만 기존의 체계에 반한 부작용 또한 만만치 않을 것이며 여러 요인들을 고려하여 OTP 도입의 문제를 최소화 시키고 기존 체제보다 큰 보안 효과를 이끌어내야 할 것이다.

목 차

<표 차례>

<그림 차례>

1. 서론

1.1. 연구의 배경 및 목적

1.1.1. 연구 목적

현재 흔히 사용하고 있는 UserID , Password를 인증기반으로 하고 있는 system은 사용자의 부주의 등으로 password를 누출시킬 경우 위험하다.
또한 TCP/IP protocol은 design 당시 보안 문제는 고려하지 않았기 때문에 sniffing과 IP spoofing 등을 이용한 해킹이 많이 일어나고 있다.
따라서 국내에서 발생한 대부분의 해킹은 ID와 password을 도용한 사례가 주류를 이루고 있기 때문에, 이에 우리 3조는 차세대 스마트폰 OS 로 떠오른 안드로이드 플랫폼을 기반으로 한 O.T.P 시스템을 구현해 봄으로써 도용으로 인한 보안상의 문제를 해결하고자한다.

1.1.2. OTP 소개 및 종류

기존의 password가 sniffing 등으로 가로 채여 보안상 도용될 위험이 있지만, OTP(One Time Password)란 말 그대로 한번 쓰고 password를 버리는 일회용 password이므로 새로 생성된 password를 사용하여 안전할 수 있다.
금융권에서 보안 카드 대신 써왔던 OTP 솔루션은 그 범위가 커져 점차 활성화되고 있다. 그러나 최근에는 금융권에서만 사용하는 것이 아니라, 고객의 개인정보보호나 기업의 기술유출 방지 시스템의 일환으로 점차 각광을 받을 전망이다.
이미 많은 게임, 포털 사이트 등에서 선택 사항으로 채택하고 있는데, 이러한 추세로 볼 때 앞으로는 기업의 이미지와 고객의 정보를 관리하기 위해서는 선택이 아닌 필수사항이 될 것이다.

(1) O.T.P 의 원리

처음 OTP를 발급받을 때 OTP기기의 종류와 시리얼 번호를 등록하게 된다. 그리고 통합인증서버에도 OTP기기에 대한 정보가 입력되고, 이를 바탕으로 인터넷 공간에서 고객인증을 하게 되는 것이다. 물론 그 사이에는 서로 다른 알고리즘이 들어가게 되는 등 복잡한 과정이 있지만, **결국에는 '자신이 가지고 있는 OTP에서 생성한 비밀번호'와 '통합인증센터에서 생성한 비밀번호'가 일치하는 지 여부를 확인하게 되는 것이다.**
이렇게 되면 사용자가 인증을 요구할 때마다 개인이 가지고 있는 OTP기기를 통해 생성된 패스워드와 인증센터 내에서 생성된 패스워드를 비교해서 응답결과를 내놓게 된다. 이 둘은 모두 정해진 것이 아니기 때문에 해커들이 이를 알아내기도 불가능하고, 알아낸다 하더라도 사용할 수 없다.

(2) O.T.P의 종류

① 토큰형 : 일반적인 OTP로 예전의 삐삐 형태를 가지고 있다. 이 안에 시간 흐름을 파악 할 수 있는 수정과 알고리즘 칩이 들어가 비밀번호를 생성한다.

② 카드형 : OTP카드는 기존 토큰형 OTP를 카드형으로 만든 제품이다.

③ **모바일형** : 모바일 OTP(MOTP)는 휴대전화에 탑재를하여 일회용 비밀번호를 생성하는S/W이다. OTP SW는 매번 다른 비밀번호를 생성해 주기 때문에 강력한 보안성을 제공받을 수 있다. 이 외에도 휴대전화에 OTP SW를 탑재함으로써 언제 어디서든 사용자 인증을 할 수 있다는 장점이 있다.
그러나 우리나라에서는 보안매체를 분리한다는 정책에 따라 현재는 게임업체를 비롯해 상대적으로 보안상의 문제가 적은 곳 위주로 사용되고 있다. 주기적으로 교체해야하는 HW 토큰(Token)과 달리 영구적인 사용이 가능한 SW 토큰 방식이기 때문에 운영비용을 절감 할 수 있다.

1.1.3. 안드로이드 모바일 플랫폼 소개

(1) 안드로이드 란?

안드로이드 (Android; 인간형 로봇)는 모습과 행동이 인간을 닮은 로봇이라는 의미이다. 원 뜻은 "인간형"이다.

(2) 안드로이드 폰[스마트 폰] ?

구글에서 제공하는 안드로이드 OS를 탑재한 스마트폰으로 아이폰은 애플OS를 사용하고 있다. 아이폰은 폐쇄형 안드로이드폰은 개방형 에플리케이션 이다.

(3) 안드로이드 OS 변천사

컵케이크(1.5) --> 도넛(1.6) --> 에클레어(2.1) --> 프로요(2.2)[현재]

(4)버젼별 특성

① 1.1 (Bender) : 윈도우에서 USB 드라이버 지원, 영어/독일어 지원

② 1.5 (Cupcake) : 한국어를 포함한 다국어 지원, 비디오 레코딩, 구글 어플 기능 개선, 브라우저 UI 업그레이드

③ 1.6 (Donut) : 카메라 기능 개선, SMS 매니저 지원, 캠코더/갤러리 개선, 자동 스크린 맞춤 기능 개선

④ 2.1 (Eclair) : 버추얼 키보드 기능 개선, 이메일 Exchange 지원, 블루투스 지원

⑤ 2.2 (Froyo) : 처리속도 향상, 에플리케이션 저장공간확대, 어도비 플레시 지원, USB 테더링과 와이파이(Wi-Fi) 핫스팟 기능을 지원

(5) 안드로이드의 스마트폰 시장 수용성

안드로이드 오픈 소스에서는 자바 VM의 속도 향상 기술인 JIT를 달빅에서도 시험 적용하고 있었다.
Google IO라고 하는 구글의 개발자 행사에서도 안드로이드 JIT에 관한 세션이 마련되어있다.

위의 Dalvik Turbo가 현재 안드로이드에서 시험하고 있는 JIT와 같은 것을 가리키는 것인지 모르겠지만, 확실한 것은 조만간 안드로이드 애플리케이션 실행속도가 대폭 향상될 것이다.

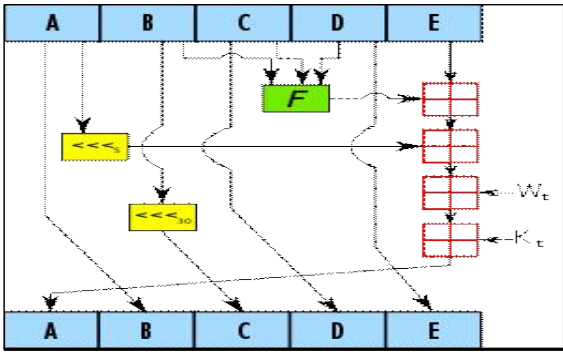
달빅VM 개선을 통한 애플리케이션의 실행속도 향상은 배터리의 절약과도 직결되는 문제이기 때문. 현재 아이폰과 더불어 수요가 급증함에 따라 개인정보유출이나 스마트폰 해킹공격이 가능해지기 때문에 OTP의 중요성도 상당해지고 있는 시점이다.

2. 이론적 배경

2.1. OTP 및 인증서버에 사용되는 암호 기술

(1) SHA-1

SHA(Secure Hash Algorithm, 안전한 해쉬 알고리즘) 함수들은 서로 관련된 암호학적 해쉬 함수들의 모음이다. 이들 함수는 미국 국가 안전 보장국(NSA)이 1993년에 처음으로 설계했으며 미국 국가 표준으로 지정되었다.



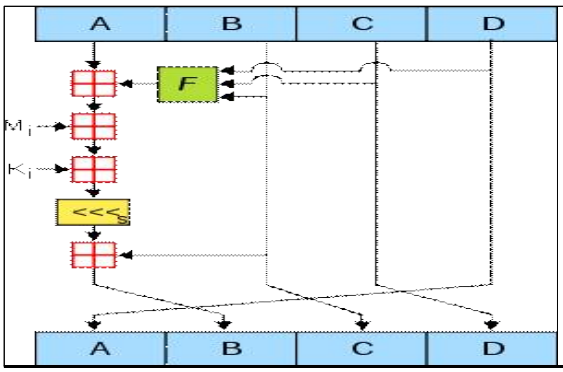
A, B, C, D, E는 각각 32비트 내부 상태이고, F는 계속 변하는 비선형 함수이며, K_t는 상수이다. $\lll_n$은 n비트만큼 왼쪽으로 회전하는 연산이고, ⊞은 법 2^{32}에 대한 덧셈을 나타낸다

<그림1 > SHA-1 압축함수가 한 블록을 처리하는 과정

(2) MD5

MD5(Message-Digest algorithm 5)는 말 그대로 메시지 축약 알고리즘으로서 128비트의 해쉬를 제공한다. RFC-1321에 정의되어 있으며, 현재는 파일 무결성 검사용으로 많이 쓰이고 있다.

MD5는 임의의 길이의 메시지(variable-length message)를 입력받아, 128비트짜리 고정 길이의 출력 값을 낸다. 입력 메시지는 512 비트 블록들로 쪼개진다. 메시지를 우선 패딩 하여 512로 나누어 떨어 질 수 있는 길이가 되게 한다. 패딩은 다음과 같이 한다. 우선 첫 단일 비트인 1을 메시지 끝부분에 추가한다. 512의 배수의 길이보다 64 비트가 적은 곳까지 0으로 채운다. 나머지 64 비트는 최초의(오리지널) 메시지의 길이를 나타내는 64 비트 정수(integer)값으로 채워진다.



MD5에서는 이 단일 연산을 64번 실행한다. 16개의 연산을 그룹화한 4 라운드로 묶인다. F는 각 라운드에서 사용하는 비선형 함수를 가리키며, 각 라운드에서는 각각 다른 함수를 사용한다. M_i는 입력 메시지의 32-비트 블록을 의미한다. $\lll_s$는 s칸 만큼을 왼쪽으로 회전을 가리키며, s는 각 연산 후 값이 변한다. ⊞은 법 2^{32} 덧셈을 말한다.

<그림2 > 단일 MD5 연산

2.2. OTP의 보안 요구사항

(1) **비밀키:** 사용자와 서버가 사전에 약속해둔 키
(2) **공유정보:** 패스워드 생성할 때 공유하는 특정 정보 (시간, 인증 횟수 등)
(3) **로그인 정보:** 비밀키 입력 전에 특정 사용자를 입증하기 위한 정보(id, pw)
(4) **알고리즘을 통해 구현한 보안기술 :** 일반적으로 OTP에서 생성해내는 비밀번호는 **특수한 알고리즘을 이용해 생성한다.** 기본적으로 **해시 알고리즘을 이용**하는데, 이것은 일방향 함수로 일정한 답을 바탕으로 그 원리를 파악할 수 없는 방식이다. **다시 말해 해시함수를 이용해 생성한 키를 해킹하는 것 자체가 불가능하다는 뜻이다.**
해시함수 자체가 원래 입력 값을 삭제하고 몇 개의 값으로 나누어, 그 번호를 역순으로 출력해서 보여주는 것을 반복하게 된다. 다음 숫자가 만들어 지면서 그 전의 자료가 계속 지워지기를 반복하기 때문에 이를 거꾸로 거슬러 문제를 푼다는 것은 불가능하다. 또한 이 함수는 비교한 후 값을 지워버리는 특징 때문에 어떤 흔적도 남지 않는다.

2.3. OTP 보안 구조

OTP 인증서버로부터 받은 질의 값을 사용자가 직접 OTP 툴에 입력하고 생성된 OTP 값을 응답 값으로 전송하여 인증하는 방식으로 서버가 전송한 challenge값을 공유 정보로 가지며 계속적인 공유정보 유지가 필요 없으며 사고 발생 시 명백한 책임 소재가 가능하다는 장점이 있다.

<그림3> Challenge Response 방식

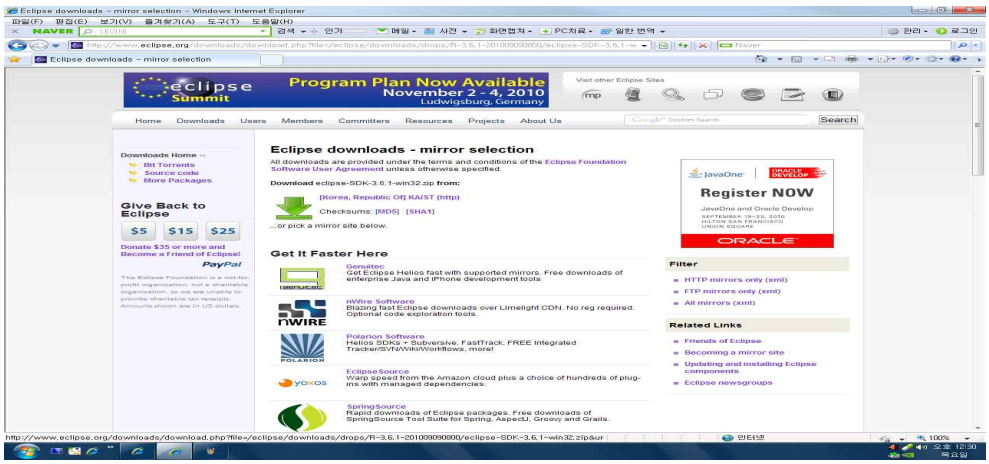
3. 연구 내용

3.1. 설계

3.1.1. OTP의 기본설계

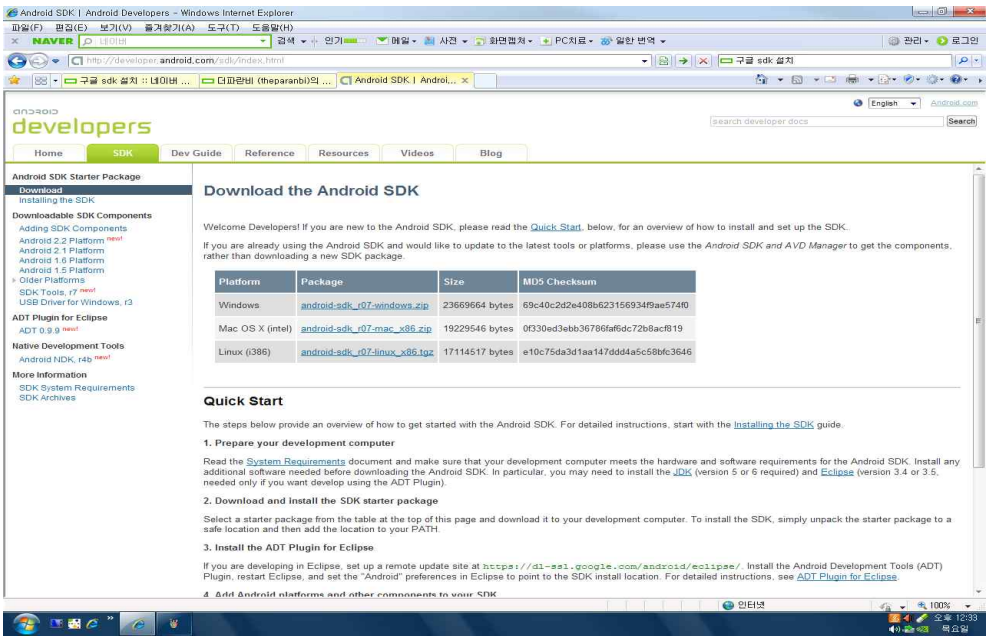
(1) 안드로이드 개발환경 구축

① 이클립스 다운로드



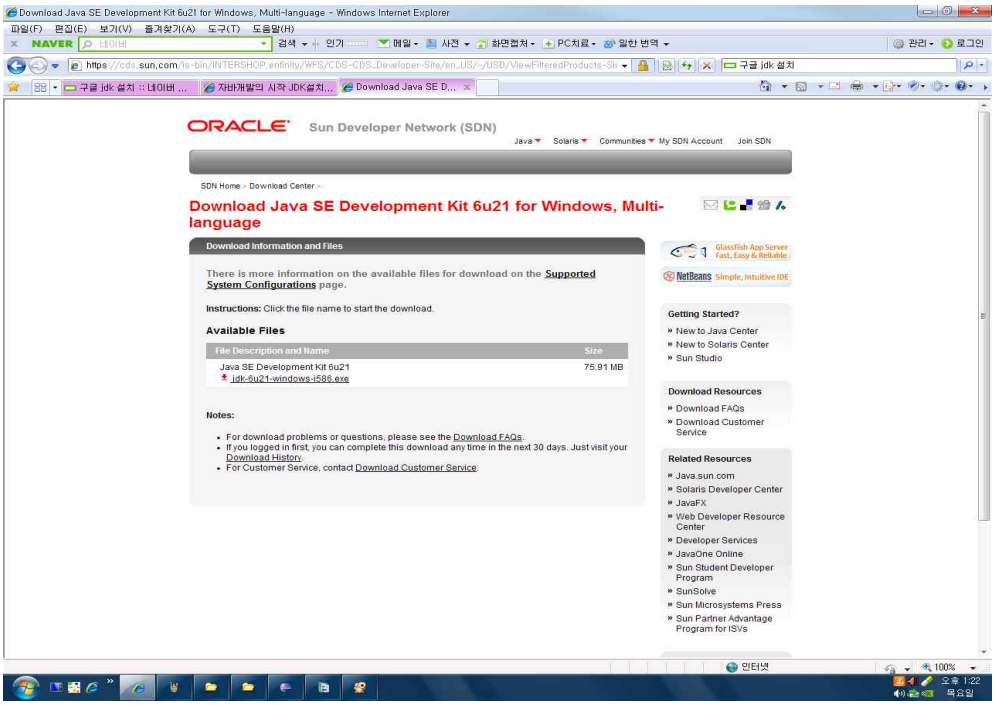
<그림4> 이클립스 다운로드 화면

② 안드로이드 SDK 설치



<그림5> 안드로이드 SDK 다운로드 화면

③ 자바 JDK 설치



<그림6> 자바 JDK 다운 화면

3.1.2. 올드보이 안드로이드 모바일 OTP Source 코드

(1) Source 코드

>kr.ac.joongbu.crypto

> GeneralDigest.java

```
package kr.ac.joongbu.crypto;
/**
 * base implementation of MD4 family style digest as outlined in
 * "Handbook of Applied Cryptography", pages 344 - 347.
 */
public abstract class GeneralDigest
{
    private static final int BYTE_LENGTH = 64;
```

```
private byte[]   xBuf
private int      xBufOff

private long     byteCount

/**
 * Standard constructor
 */
protected GeneralDigest()
{
    xBuf = new byte[4];
    xBufOff = 0;
}

/**
 * Copy constructor.  We are using copy constructors in place
 * of the Object.clone() interface as this interface is not
 * supported by J2ME.
 */
protected GeneralDigest(GeneralDigest t)
{
    xBuf = new byte[t.xBuf.length];
    System.arraycopy(t.xBuf, 0, xBuf, 0, t.xBuf.length);

    xBufOff = t.xBufOff
    byteCount = t.byteCount
}

public void update(
    byte in)
{
    xBuf[xBufOff++] = in;

    if (xBufOff == xBuf.length)
    {
        processWord(xBuf, 0);
        xBufOff = 0;
    }
```

```
    byteCount++;
}

public void update(
    byte[]  in,
    int     inOff,
    int     len)
{
    //
    // fill the current word
    //
    while ((xBufOff != 0) && (len > 0))
    {
        update(in[inOff]);

        inOff++;
        len--;
    }

    //
    // process whole words.
    //
    while (len > xBuf.length)
    {
        processWord(in, inOff);

        inOff += xBuf.length
        len -= xBuf.length
        byteCount += xBuf.length
    }

    //
    // load in the remainder.
    //
    while (len > 0)
    {
        update(in[inOff]);

        inOff++;
```

```
            len--;
        }
    }

    public void finish()
    {
        long      bitLength = (byteCount << 3);

        //
        // add the pad bytes.
        //
        update((byte)128);

        while (xBufOff != 0)
        {
            update((byte)0);
        }

        processLength(bitLength);

        processBlock();
    }

    public void reset()
    {
        byteCount = 0;

        xBufOff = 0;
        for (int i = 0; i < xBuf.length i++)
        {
            xBuf[i] = 0;
        }
    }
    public int getByteLength()
    {
        return BYTE_LENGTH
    }

    protected abstract void processWord(byte[] in, int inOff);
```

```
    protected abstract void processLength(long bitLength);
    protected abstract void processBlock();
}
```

〉Pack.java

```
package kr.ac.joongbu.crypto;


public abstract class Pack
{
    public static int bigEndianToInt(byte[] bs, int off)
    {
        int n = bs[  off] << 24;
        n |= (bs[++off] & 0xff) << 16;
        n |= (bs[++off] & 0xff) << 8;
        n |= (bs[++off] & 0xff);
        return n;
    }

    public static void intToBigEndian(int n, byte[] bs, int off)
    {
        bs[  off] = (byte)(n >>> 24);
        bs[++off] = (byte)(n >>> 16);
        bs[++off] = (byte)(n >>>  8);
        bs[++off] = (byte)(n         );
    }

    public static long bigEndianToLong(byte[] bs, int off)
    {
        int hi = bigEndianToInt(bs, off);
        int lo = bigEndianToInt(bs, off + 4);
        return ((long)(hi & 0xffffffffL) << 32) | (long)(lo & 0xffffffffL);
    }

    public static void longToBigEndian(long n, byte[] bs, int off)
    {
```

```
        intToBigEndian((int)(n >>> 32), bs, off);
        intToBigEndian((int)(n & 0xffffffffL), bs, off + 4);
    }
}
```

〉SHA1Digest.java

```
package kr.ac.joongbu.crypto;

/**
 * implementation of SHA-1 as outlined in "Handbook of Applied Cryptography", pages 346 - 349.
 *
 * It is interesting to ponder why the, apart from the extra IV, the other difference here from MD5
 * is the "endienness" of the word processing!
 */
public class SHA1Digest
    extends GeneralDigest
{
    private static final int     DIGEST_LENGTH = 20;

    private int       H1, H2, H3, H4, H5

    private int[]    X = new int[80];
    private int       xOff

    /**
     * Standard constructor
     */
    public SHA1Digest()
    {
        reset();
    }

    /**
     * Copy constructor.  This will copy the state of the provided
     * message digest.
     */
    public SHA1Digest(SHA1Digest t)
    {
```

```
        super(t);

        H1 = t.H1
        H2 = t.H2
        H3 = t.H3
        H4 = t.H4
        H5 = t.H5

        System.arraycopy(t.X, 0, X, 0, t.X.length);
        xOff = t.xOff
    }

    public String getAlgorithmName()
    {
        return "SHA-1"
    }
    public int getDigestSize()
    {
        return DIGEST_LENGTH
    }

    protected void processWord(
        byte[]  in,
        int     inOff)
    {
        // Note: Inlined for performance
//          X[xOff] = Pack.bigEndianToInt(in, inOff);
        int n = in[  inOff] << 24;
        n |= (in[++inOff] & 0xff) << 16;
        n |= (in[++inOff] & 0xff) << 8;
        n |= (in[++inOff] & 0xff);
        X[xOff] = n;

        if (++xOff == 16)
        {
            processBlock();
        }
    }
    protected void processLength(
```

```
    long      bitLength)
{
    if (xOff > 14)
    {
        processBlock();
    }

    X[14] = (int)(bitLength >>> 32);
    X[15] = (int)(bitLength & 0xffffffff);
}

public int doFinal(
    byte[]  out,
    int     outOff)
{
    finish();

    Pack.intToBigEndian(H1, out, outOff);
    Pack.intToBigEndian(H2, out, outOff + 4);
    Pack.intToBigEndian(H3, out, outOff + 8);
    Pack.intToBigEndian(H4, out, outOff + 12);
    Pack.intToBigEndian(H5, out, outOff + 16);

    reset();

    return DIGEST_LENGTH
}

/**
 * reset the chaining variables
 */
public void reset()
{
    super.reset();

    H1 = 0x67452301;
    H2 = 0xefcdab89;
    H3 = 0x98badcfe;
    H4 = 0x10325476;
```

```
    H5 = 0xc3d2e1f0;

    xOff = 0;
    for (int i = 0; i != X.length i++)
    {
        X[i] = 0;
    }
}

//
// Additive constants
//
private static final int    Y1 = 0x5a827999;
private static final int    Y2 = 0x6ed9eba1;
private static final int    Y3 = 0x8f1bbcdc;
private static final int    Y4 = 0xca62c1d6;

private int f(
    int     u,
    int     v,
    int     w)
{
    return ((u & v) | ((~u) & w));
}

private int h(
    int     u,
    int     v,
    int     w)
{
    return (u ^ v ^ w);
}

private int g(
    int     u,
    int     v,
    int     w)
{
    return ((u & v) | (u & w) | (v & w));
```

```
}

protected void processBlock()
{
    //
    // expand 16 word block into 80 word block.
    //
    for (int i = 16; i < 80; i++)
    {
        int t = X[i - 3] ^ X[i - 8] ^ X[i - 14] ^ X[i - 16];
        X[i] = t << 1 | t >>> 31;
    }

    //
    // set up working variables.
    //
    int     A = H1
    int     B = H2
    int     C = H3
    int     D = H4
    int     E = H5

    //
    // round 1
    //
    int idx = 0;

    for (int j = 0; j < 4; j++)
    {
        // E = rotateLeft(A, 5) + f(B, C, D) + E + X[idx++] + Y1
        // B = rotateLeft(B, 30)
        E += (A << 5 | A >>> 27) + f(B, C, D) + X[idx++] + Y1
        B = B << 30 | B >>> 2;

        D += (E << 5 | E >>> 27) + f(A, B, C) + X[idx++] + Y1
        A = A << 30 | A >>> 2;

        C += (D << 5 | D >>> 27) + f(E, A, B) + X[idx++] + Y1
        E = E << 30 | E >>> 2;
```

```
    B += (C << 5 | C >>> 27) + f(D, E, A) + X[idx++] + Y1
    D = D << 30 | D >>> 2;

    A += (B << 5 | B >>> 27) + f(C, D, E) + X[idx++] + Y1
    C = C << 30 | C >>> 2;
}

//
// round 2
//
for (int j = 0; j < 4; j++)
{
    // E = rotateLeft(A, 5) + h(B, C, D) + E + X[idx++] + Y2
    // B = rotateLeft(B, 30)
    E += (A << 5 | A >>> 27) + h(B, C, D) + X[idx++] + Y2
    B = B << 30 | B >>> 2;

    D += (E << 5 | E >>> 27) + h(A, B, C) + X[idx++] + Y2
    A = A << 30 | A >>> 2;

    C += (D << 5 | D >>> 27) + h(E, A, B) + X[idx++] + Y2
    E = E << 30 | E >>> 2;

    B += (C << 5 | C >>> 27) + h(D, E, A) + X[idx++] + Y2
    D = D << 30 | D >>> 2;

    A += (B << 5 | B >>> 27) + h(C, D, E) + X[idx++] + Y2
    C = C << 30 | C >>> 2;
}
//
// round 3
//
for (int j = 0; j < 4; j++)
{
    // E = rotateLeft(A, 5) + g(B, C, D) + E + X[idx++] + Y3
    // B = rotateLeft(B, 30)
    E += (A << 5 | A >>> 27) + g(B, C, D) + X[idx++] + Y3
    B = B << 30 | B >>> 2;
```

```
    D += (E << 5 | E >>> 27) + g(A, B, C) + X[idx++] + Y3
    A = A << 30 | A >>> 2;

    C += (D << 5 | D >>> 27) + g(E, A, B) + X[idx++] + Y3
    E = E << 30 | E >>> 2;

    B += (C << 5 | C >>> 27) + g(D, E, A) + X[idx++] + Y3
    D = D << 30 | D >>> 2;

    A += (B << 5 | B >>> 27) + g(C, D, E) + X[idx++] + Y3
    C = C << 30 | C >>> 2;
}

//
// round 4
//
for (int j = 0; j <= 3; j++)
{
    // E = rotateLeft(A, 5) + h(B, C, D) + E + X[idx++] + Y4
    // B = rotateLeft(B, 30)
    E += (A << 5 | A >>> 27) + h(B, C, D) + X[idx++] + Y4
    B = B << 30 | B >>> 2;

    D += (E << 5 | E >>> 27) + h(A, B, C) + X[idx++] + Y4
    A = A << 30 | A >>> 2;

    C += (D << 5 | D >>> 27) + h(E, A, B) + X[idx++] + Y4
    E = E << 30 | E >>> 2;
    B += (C << 5 | C >>> 27) + h(D, E, A) + X[idx++] + Y4
    D = D << 30 | D >>> 2;
    A += (B << 5 | B >>> 27) + h(C, D, E) + X[idx++] + Y4
    C = C << 30 | C >>> 2;
}


H1 += A;
H2 += B;
H3 += C;
H4 += D;
H5 += E;
```

```
        //
        // reset start of the buffer.
        //
        xOff = 0;
        for (int i = 0; i < 16; i++)
        {
            X[i] = 0;
        }
    }
}
```

>kr.ac.joongbu.motp

>Mobile.OTP.java

```
package kr.ac.joongbu.motp;
import org.bouncycastle.util.encoders.Hex;
import kr.ac.joongbu.crypto.SHA1Digest;
import android.app.Activity;
import android.os.Bundle;
import android.text.Editable;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.EditText;
import android.widget.TextView;
public class MobileOTP extends Activity implements OnClickListener {
        private static byte[] key = new byte[] {'a','b','c','d'};
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // 버튼에 리스너 설정
        View createButton = this.findViewById(R.id.CreateOTPButton);
        createButton.setOnClickListener(this);
    }
        public void onClick(View v) {
```

```
                switch (v.getId()) {
                case R.id.CreateOTPButton:
                                TextView otpNumberView = (TextView)
this.findViewById(R.id.OTPNumberTextView);
                                EditText inputChallenge = (EditText)
this.findViewById(R.id.ChallengeEditText);
                Editable text = inputChallenge.getText();
                String optString = calcOTP(text.toString());
                otpNumberView.setText(optString);
                        break;
                }

        }
                public String calcOTP(String challenge) {
                // SHA256 해시값 저장 변수 output
                byte[] hashOutput = new byte[20];

                // String 형태의 challenge를 byte형태로 변환
                byte[] c = challenge.getBytes();
                // output  = SHA256(key || c)
                SHA1Digest digest = new SHA1Digest();
                digest.update(key, 0, key.length);
                digest.update(c, 0, c.length);
                digest.doFinal(hashOutput, 0);

                // 해시값 hashOutput을 숫자로 변환
                String cc = new String(Hex.encode(key));
                cc += new String(Hex.encode(c));
                String hashString = new String(Hex.encode(hashOutput));
                String otp = convertHashOutputToOTPString(hashString.getBytes());

                return otp;
        }
                public String convertHashOutputToOTPString(byte[] hashOutput) {
                        int number = 0;

                // 해시값 2바이트를 숫자로 변환
                char a = (char)hashOutput[0];
                number += (char)hashOutput[0];
```

```
                number += ((char)hashOutput[1])*256;
                //number += hashOutput[2]*256*256;


                // 음수이면 양수로 변환
                if (number < 0)
                        number *= -1;

                // 결과값(number)를 문자열로 변환
                String otpString = new String(""+number);

                return otpString;
        }
}
```

(2) gen

>kr.ac.joongbu.motp

> R.java

```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found.  It
 * should not be modified by hand.
 */

package kr.ac.joongbu.motp;
public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class id {
        public static final int ChallengeEditText=0x7f050002;
        public static final int CreateOTPButton=0x7f050003;
        public static final int LinearLayout01=0x7f050000;
        public static final int OTPNumberTextView=0x7f050004;
        public static final int TextView01=0x7f050001;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040000;
    }
}
```

(3) res

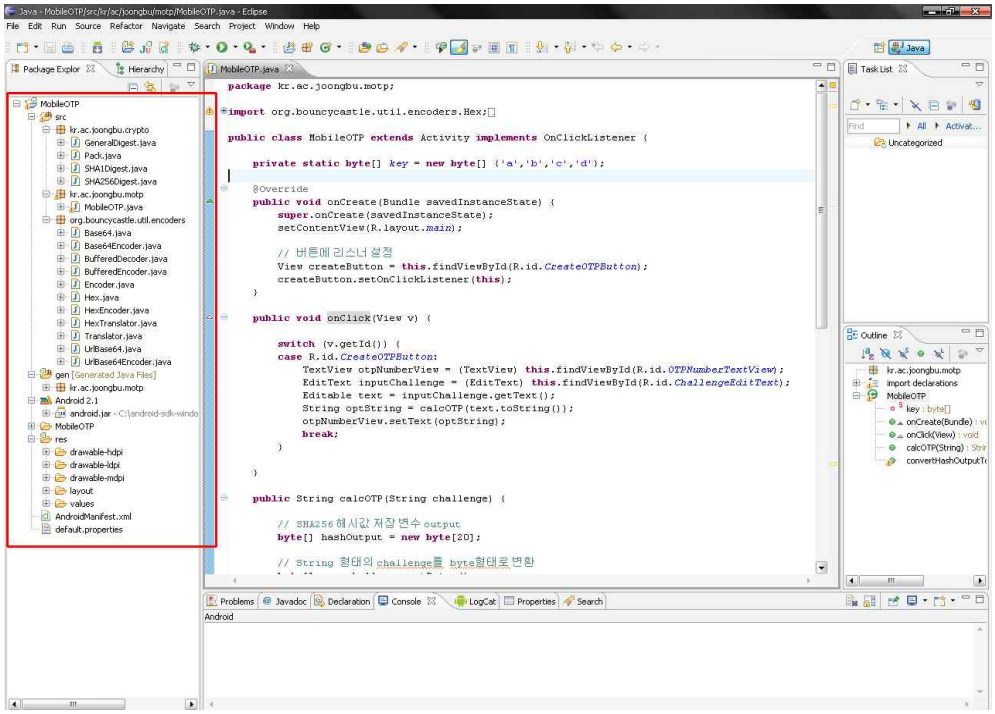
>layout

> main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="30px">
```

```
<LinearLayout android:id="@+id/LinearLayout01" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:paddingBottom ="30px">
</LinearLayout>

<TextView android:text="PC
android:id="@+id/TextView01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:paddingBottom ="10px"
android:textSize="12px">
</TextView>

<EditText android:id="@+id/ChallengeEditText"
android:layout_width="200px"
android:layout_height="wrap_content"
android:layout_gravity="center" android:text="">
</EditText>

<LinearLayout android:id="@+id/LinearLayout01" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:paddingBottom ="30px">
</LinearLayout>

<Button android:text="OTP
android:id="@+id/CreateOTPButton"
android:layout_width="200px"
android:layout_height="wrap_content"
android:layout_gravity="center"></Button>
<LinearLayout android:id="@+id/LinearLayout01" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:paddingBottom ="60px">
</LinearLayout>

<TextView android:text="OTP"
android:id="@+id/TextView01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:textSize="10px">
</TextView>

<TextView android:text="000000"
android:id="@+id/OTPNumberTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:paddingBottom ="50px"
android:textSize="36px">
</TextView>

</LinearLayout>
```

3.1.3. 이클립스 작업 환경



<그림7> 이클립스 작업환경(구글 API SDK 라이브러리)

<그림8> 이클립스 에뮬레이터 테스팅 (초기화면)

<그림9> 이클립스 에뮬레이터 테스팅 (메뉴화면)

<그림10> 이클립스 에뮬레이터 테스팅 (OTP 화면))

3.1.4. 올드보이 OTP 인증 서버

(1) 서버 설치 장소 : C5-415에 위치한 PC
Windows Server 2003 Sp2
3 Team URL : isotp.joongbu.ac.kr

(2) IP 주소
IP : 61.81.108.24
Subnet Mask : 255.255.255.0
Default Gateway : 61.81.108.1

(3) PHP 작업 환경
PHP myadmin 127.0.0.1
MySQL UTF-8 Unicode
Ver 5.1.41-community Web Server
Apache 2.2.14 PHP 5.2.12
MySQL Client Version 5.0.51a
PHP extension: mysql

(4) 데이터베이스 구조

필드 기능	필드 이름	속성	기본값	기타
레코드고유 번호	rowid	int(11)		필수, 기본키, 자동증가
회원 아이디	userid	char(20)		인덱스
암호	password	char(100)		
이름	name	char(50)		
우편번호	zip	char(10)		
주소	address	char(255)		
연락처	tel	char(500		
이메일	email	char(100)		
회원 등급	user_level	int(11)	0	

<표1> 모바일 인증서버 회원가입 데이터베이스 user 테이블 구조

(5) user 테이블의 SQL 스크립트

```
CREATE TABLE IF NOT EXISTS user (
int(11) NOT NULL auto_increment,
```

```
  userid char(20) default NULL,
  password chrar(100) default NULL,
  name char(50) default NULL,
  address char(255) default NULL,
  tel char(50) default NULL,
  email char(100) default NULL,
  zip char(10) default NULL,
  user_level int(11) default '0',

  PRIMARY KEY (rowid),
  KEY userid (userid)
);
```

3.1.5. 데이터베이스 구성

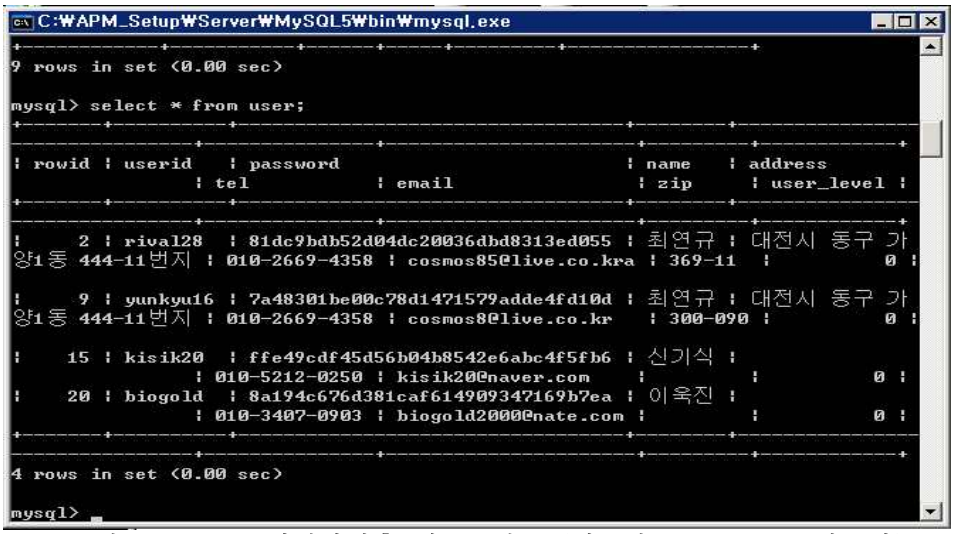
(1) 3team : 메인에서 O.T.P 인증을 위한 3team 자체 독자적 데이터베이스
gnuboard_db : 인증후 웹페이지에서 일부 사용할 그누보드 데이터베이스

```
C:₩APM_Setup₩Server₩MySQL5₩bin₩mysql.exe
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 198
Server version: 5.1.41-community MySQL Community Server (GPL)

Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| 3team              | O.T.P 인증 데이터베이스
| gnuboard_db        | 인증후 웹페이지 데이터베이스(그누보드)
| mydb               |
| mysql              |
| phpmyadmin         |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```

<그림11> 데이터베이스 구성

```
C:₩APM_Setup₩Server₩MySQL5₩bin₩mysql.exe
mysql> show tables;
+-----------------+
| Tables_in_3team |
+-----------------+
| user            |
+-----------------+
1 row in set (0.00 sec)

mysql> desc user;
+-------------+-----------+------+-----+---------+----------------+
| Field       | Type      | Null | Key | Default | Extra          |
+-------------+-----------+------+-----+---------+----------------+
| rowid       | int(11)   | NO   | PRI | NULL    | auto_increment |
| userid      | char(20)  | YES  | MUL | NULL    |                |
| password    | char(100) | YES  |     | NULL    |                |
| name        | char(50)  | YES  |     | NULL    |                |
| address     | char(255) | YES  |     | NULL    |                |
| tel         | char(50)  | YES  |     | NULL    |                |
| email       | char(100) | YES  |     | NULL    |                |
| zip         | char(10)  | YES  |     | NULL    |                |
| user_level  | int(11)   | YES  |     | 0       |                |
+-------------+-----------+------+-----+---------+----------------+
9 rows in set (0.00 sec)

mysql>
```

<그림12 -3team 데이터베이스 user 테이블>



<그림13> 3team 회원가입후 레코드값들 (패스워드는 MD5로 암호화)

(2) 회원 가입 폼 출력 파일

└ http://61.81.108.24/user/add_form.php

루트 index.php - main.php 파일 인클루드

<그림14> 인터넷 플렛폼

/새창으로 add_form.php 열기

<area shape="rect" coords="76, 11, 193, 35" href="javascript:na_open_window('회원가입', 'http://61.81.108.24/user/add_form.php', 0, 0, 510, 470, 0, 0, 0, 0, 0)" target="_self">

3.1.6. USER 폴더

(1) add_form.php

```
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<link href="../css/global.css" rel="stylesheet" type="text/css">
</head>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<br>
<br>
<div align="left">
    <table width="500" border="0" cellpadding="5" cellspacing="0" bgcolor="#EBDBF2"
style="border:0px #f9253b solid;border-top-width:3px;">
        <tr>
            <td bgcolor="white"><b>회원가입 폼</b></td>
            <td align="right" bgcolor="white"> </td>
        </tr>
    </table>
</div>
<br>
<div align="left">
    <table width="500" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td align="center">정확히 입력하신 후 등록하세요! &quot;<font
 color="#FF0000">*</font>&quot;는 필수 입력사항입니다.</td>
        </tr>
    </table>
</div>
<br>
<form action="add.php" method="post" name="user_form" id="user_form" style="margin0px;" onSubmit="return checkForm(this)">
    <div align="left">
        <table width="500" border="0" cellpadding="5" cellspacing="1">
            <tr>
                <td width="100" bgcolor="white"><font color="#FF0000">*</font> 아이디</td>
                <td bgcolor="white"> <input name="userid" type="text" id="userid"
```

```
size="20" onblur="if (this.value!='') checkId(this.form);">
        <input type="button" name="Button" value="중복검사" onclick="checkId(this.form);">
                <script>
            function checkId(theForm) {
                    if (theForm.userid.value=='') {
                            alert('아이디를 입력하십시오.');
                            theForm.userid.focus();
                    } else {
                            var                checkIdWin                =
window.open('checkid.php?userid='+theForm.userid.value,'','width=300,height=200');
                            checkIdWin.focus();
                    }
            }
                            </script>
            </td>
        </tr>
        <tr>
            <td bgcolor="white"><font color="#FF0000">*</font> 암호</td>
            <td bgcolor="white"> <input name="password" type="password" id="password" size="20">
            </td>
        </tr>
        <tr>
            <td bgcolor="white"><font color="#FF0000">*</font> 암호확인</td>
            <td bgcolor="white"> <input name="password_re" type="password" id="password_re" size="20">
            </td>
        </tr>
        <tr>
            <td bgcolor="white"><font color="#FF0000">*</font> 이름</td>
            <td bgcolor="white"> <input name="name" type="text" id="name" size="20">
        예)홍길동  </td>
        </tr>
        <tr>
            <td bgcolor="white"><font color="#FF0000">*</font> 이메일</td>
            <td bgcolor="white"> <input name="email" type="text" id="email" size="30">
        예)userid@domain.com</td>
        </tr>
        <tr>
            <td bgcolor="white"><font color="#FF0000">*</font> 연락처</td>
            <td bgcolor="white"> <input name="tel" type="text" id="tel" size="20">
```

```
        예)02-000-0000</td>
            </tr>
            <tr>
                <td bgcolor="white">   우편번호</td>
                <td bgcolor="white"> <input name="zip" type="text" id="zip" size="7">
        예)135-080  </td>
            </tr>
            <tr>
                <td bgcolor="white">   주소</td>
                <td bgcolor="white"> <input name="address" type="text" id="address" size="50">
                </td>
            </tr>
        </table>
    </div>
  <br>
    <div align="left">
        <table width="500" height="40" border="0" cellpadding="0" cellspacing="0"
bgcolor="#EBDBF2" style="border:0px #f9253b solid;border-bottom-width:3px;">
            <tr>
                <td align="center" bgcolor="white"> <input type="submit" name="Submit" value="등록합니다."> 
<input type="reset" name="Reset" value="취소">
                </td>
            </tr>
        </table>
    </div>
</form>
<script>
function checkForm(theForm) {
        if (theForm.userid.value==="") {
                alert("아이디를 입력하십시오.");
                theForm.userid.focus();
                return false;
        } else if (theForm.password.value==="") {
                alert("암호를 입력하십시오.");
                theForm.password.focus();
                return false;
        } else if (theForm.password_re.value==="") {
                alert("암호확인을 입력하십시오.");
                theForm.password_re.focus();
```

```
                return false;
        } else if (theForm.password.value!=theForm.password_re.value) {
                alert("암호를 잘못입력하셨습니다.\r\n\r\n다시 입력하십시오.");
                theForm.password.value="";
                theForm.password_re.value="";
                theForm.password.focus();
                return false;
        } else if (theForm.name.value=="") {
                alert("이름을 입력하십시오.");
                theForm.name.focus();
                return false;
        } else if (theForm.email.value=="") {
                alert("이메일을 입력하십시오.");
                theForm.email.focus();
                return false;
        } else if (theForm.tel.value=="") {
                alert("연락처를 입력하십시오.");
                theForm.tel.focus();
                return false;
        } else {
                return true;
        }

}
</script>
</body>
</html>
```

(2) add.php

```
<?
include_once ("../3Teamlib/dbcon.php");

//이미 등록된 회원인지를 검사합니다.
$sql = "select count(*) from user where userid='$userid' ";
$res = mysql_query($sql);
$rs = mysql_fetch_row($res);
$already_registered_count = $rs[0];
```

```
if ($already_registered_count>0) {
        echo "
        <script>
        alert('[아이디 중복] 이미 등록된 아이디입니다.\\r\\n\\r\\n다른 아이디로 등록하십시오.');
        history.back();
        </script>
        ";
        die;
}

//필수입력항목을 모두 입력했는지 검사합니다.
if ($userid=="" || $password=="" || $password_re=="" || $password!=$password_re ||
$name=="" || $email=="" || $tel=="") {
        echo "
        <script>
        alert('[데이터 누락] 필수입력란을 정확히 입력하십시오.');
        history.back();
        </script>
        ";
        die;
}

//데이터베이스에 등록합니다.
$sql = "
        insert into user (
        userid,password,name,email,tel,zip,address
        ) values (
        '$userid',MD5('$password'),'$name','$email','$tel','$zip','$address'
        )
                ";
$res = mysql_query($sql);
$affected_rows = mysql_affected_rows();
if ($affected_rows>0) {
        echo "
        <script>
        alert('[등록성공] 회원으로 등록되었습니다.');
        window.close();
        </script>
        ";
```

```
} else {
        echo "
        <script>
        alert('[등록실패] 데이터베이스서버의 오류 또는 회원필드 오류로 인하여 등록실
패하였습니다.');
        history.back();
        </script>
        ";
}
?>
```

(3) change_password.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

if ($session_userid=="") {
        echo "
        <script>
        alert('[회원전용] 로그인 후 사용하십시오.');
        location.replace('../login/login_form.php');
        </script>
        ";
        die;
}

//데이터베이스에 등록합니다.
$sql = "update user set password=MD5('$password') where userid='$session_userid'";
$res = mysql_query($sql);
$affected_rows = mysql_affected_rows();
if ($affected_rows>0) {
        echo "
        <script>
        alert('[암호변경성공] 새암호로 변경했습니다.');
        location.replace('../login/login_success.php');
        window.close();
        </script>
        ";
```

```
} else {
        echo "
        <script>
        alert('[암호변경실패] 데이터베이스서버의 오류 또는 회원필드 오류로 인하여 변경실패하였습니다.');
        location.back();
        </script>
        ";
}
?>
```

(4) change_password_form.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

if ($session_userid=="") {
        echo "
        <script>
        alert('[회원전용] 로그인 후 사용하십시오.');
        location.replace('../login/login_form.php');
        </script>
        ";
        die;
}

$sql = "select * from user where userid='$session_userid'";
$res = mysql_query($sql) or die("데이터베이스 오류!!");
$rs = mysql_fetch_array($res);
?>
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<link href="../css/global.css" rel="stylesheet" type="text/css">
</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<br>
```

```
<br>
<div align="left">
    <table width="500" border="0" cellpadding="5" cellspacing="0" bgcolor="#EBDBF2"
style="border:0px #f9253b solid;border-top-width:3px;">
        <tr>
            <td bgcolor="white"><b>암호변경</b></td>
            <td align="right" bgcolor="white"> </td>
        </tr>
    </table>
</div>
<br>
<div align="left">
    <table width="500" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td align="center">변경할 암호를 정확히 입력하신 후 변경버튼을 클릭하십시오.</td>
        </tr>
    </table>
</div>
<br>
<form action="change_password.php" method="post" name="user_form" id="user_form" style="margin:0px;">
    <div align="left">
        <table width="500" border="0" cellpadding="5" cellspacing="1">
            <tr>
                <td width="100" bgcolor="white">    아이디</td>
                <td bgcolor="white">  <? echo $session_userid;?>  </td>
            </tr>
            <tr>
                <td bgcolor="white"><font color="#FF0000">*</font> 새 암호</td>
                <td bgcolor="white"> <input name="password" type="password" id="password">
                </td>
            </tr>
            <tr>
                <td bgcolor="white"><font color="#FF0000">*</font> 새 암호 확인
</td>
                <td bgcolor="white"> <input name="password_re" type="password" id="password_re">
                </td>
            </tr>
        </table>
    </div>
```

```
  <br>
    <div align="left">
        <table width="500" height="40" border="0" cellpadding="0" cellspacing="0"
bgcolor="#EBDBF2" style="border:0px #f9253b solid;border-bottom-width:3px;">
            <tr>
                <td align="center" bgcolor="white">
<input type="button" name="Button" value="변경" onClick="checkForm(this.form);">
        <input type="reset" name="Reset" value="취소">
                </td>
            </tr>
        </table>
    </div>
</form>
<script>
function checkForm(theForm) {
        if (theForm.password.value!=theForm.password_re.value) {
                alert("새암호를 잘못 입력하셨습니다.");
                theForm.password.value="";
                theForm.password_re.value="";
                theForm.password.focus();
        } else {
                theForm.submit();
        }
}
</script>
</body>
</html>
```

(5) checkid.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

$sql = "select count(*) from user where userid='$userid'";
$res = mysql_query($sql);
$rs = mysql_fetch_array($res);
$exist_num = $rs[0];
?>
```

```
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<link href="../css/global.css" rel="stylesheet" type="text/css">
</head>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<br>
<table width="280" border="0" align="center" cellpadding="5" cellspacing="0"
bgcolor="#EBDBF2" style="border:0px #f9253b solid;border-top-width:3px;">
  <tr>
    <td bgcolor="white"><b>중복확인</b></td>
    <td align="right" bgcolor="white"> </td>
  </tr>
</table>
<form method="post" name="checkid_form" id="checkid_form" style="margin:0px;">
  <table width="280" height="100" border="0" align="center" cellpadding="5"
cellspacing="1">
    <tr>
      <td align="center" bgcolor="white">
        <? if ($exist_num>0) { ?>
        &quot;<? echo $userid;?>&quot;는 이미 사용중입니다. <br> <br>
        다른 아이디를 사용하십시오.
        <? } else {?>
        &quot;<? echo $userid;?>&quot;를 사용해도 좋습니다.
        <? }//if?>
      </td>
    </tr>
  </table>
  <table width="280" height="40" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#EBDBF2" style="border:0px #f9253b solid;border-bottom-width:3px;">
    <tr>
      <td align="center" bgcolor="white"><input type="button" name="Button" value="닫기" onClick="self.close();">
      </td>
    </tr>
  </table>
</form>
</body>
</html>
```

(6) edit.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

if ($session_userid=="") {
        echo "
        <script>
        alert('[회원전용] 로그인 후 사용하십시오.');
        location.replace('../login/login_form.php');
        </script>
        ";
        die;
}

//필수입력항목을 모두 입력했는지 검사합니다.
if ($name=="" || $email="" || $tel="") {
        echo "
        <script>
        alert('[데이터 누락] 필수입력란을 정확히 입력하십시오.');
        history.back();
        </script>
        ";
        die;
}

//데이터베이스에 등록합니다.
$sql = "
        update user set
        name='$name',email='$email',tel='$tel',zip='$zip',address='$address'
        where userid='$session_userid'
        ";
$res = mysql_query($sql);
$affected_rows = mysql_affected_rows();
if ($affected_rows>0) {
        echo "
        <script>
        alert('[수정성공] 수정했습니다.');
```

```
            location.replace('edit_form.php');
            window.close();
            </script>
            ";
} else {
            echo "
            <script>
            alert('변경된 사항이 없습니다.');
            location.replace('edit_form.php');
            </script>
            ";
}
?>
```

(7) edit_form.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

if ($session_userid=="") {
            echo "
            <script>
            alert('[회원전용] 로그인 후 사용하십시오.');
            location.replace('../login/login_form.php');
            </script>
            ";
            die;
}

$sql = "select * from user where userid='$session_userid'";
$res = mysql_query($sql) or die("데이터베이스 오류!!");
$rs = mysql_fetch_array($res);
?>
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<link href="../css/global.css" rel="stylesheet" type="text/css">
</head>
```

```
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<br>
<br>
<div align="left">
    <table width="500" border="0" cellpadding="5" cellspacing="0" bgcolor="#EBDBF2"
style="border:0px #f9253b solid;border-top-width:3px;">
        <tr>
            <td bgcolor="white"><b>회원정보수정 폼</b></td>
            <td align="right" bgcolor="white"> </td>
        </tr>
    </table>
</div>
<br>
<div align="left">
    <table width="500" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td align="center">정확히 입력하신 후 등록하세요! &quot;<font color="#FF0000">*</font>&quot;는
필수 입력사항입니다.</td>
        </tr>
    </table>
</div>
<br>
<form action="edit.php" method="post" name="user_form" id="user_form" style="margin:0px;">
    <div align="left">
        <table width="500" border="0" cellpadding="5" cellspacing="1">
            <tr>
                <td width="100" bgcolor="white">    아이디</td>
                <td bgcolor="white">  <? echo $session_userid;?>  </td>
            </tr>
            <tr>
                <td bgcolor="white"><font color="#FF0000">*</font> 이름</td>
                <td bgcolor="white">  <input name="name" type="text" id="name"
value="<? echo $rs[name];?>" size="20">
        예)홍길동  </td>
            </tr>
            <tr>
                <td bgcolor="white"><font color="#FF0000">*</font> 이메일</td>
                <td bgcolor="white"> <input name="email" type="text" id="email" value="<?
echo $rs[email];?>" size="30">
        예)userid@domain.com</td>
```

```
            </tr>
            <tr>
                <td bgcolor="white"><font color="#FF0000">*</font> 연락처</td>
                <td bgcolor="white"> <input name="tel" type="text" id="tel" value="<? echo $rs[tel];?>" size="20">
        예)02-000-0000</td>
            </tr>
            <tr>
                <td bgcolor="white">   우편번호</td>
                <td bgcolor="white"> <input name="zip" type="text" id="zip" value="<? echo $rs[zip];?>" size="7">
        예)135-080  </td>
            </tr>
            <tr>
                <td bgcolor="white">   주소</td>
                <td bgcolor="white"> <input name="address" type="text" id="address" value="<? echo $rs[address];?>" size="50">
                </td>
            </tr>
        </table>
    </div>
  <br>
    <div align="left">
        <table width="500" height="40" border="0" cellpadding="0" cellspacing="0" bgcolor="#EBDBF2" style="border:0px #f9253b solid;border-bottom-width:3px;">
            <tr>
                <td align="center" bgcolor="white">
<input type="button" name="Button" value="수정합니다." onClick="checkForm(this.form);">
        <input type="reset" name="Reset" value="취소">
                </td>
            </tr>
        </table>
    </div>
</form>
<script>
function checkForm(theForm) {
        if (theForm.name.value==="") {
                alert("이름을 입력하십시오.");
                theForm.name.focus();
```

```
		} else if (theForm.email.value==="") {
			alert("이메일을 입력하십시오.");
			theForm.email.focus();
		} else if (theForm.tel.value==="") {
			alert("연락처를 입력하십시오.");
			theForm.tel.focus();
		} else {
			theForm.submit();
		}
}
</script>
</body>
</html>
```

(8) login_sucess.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

$sql = "select * from user where userid='$session_userid'";
$res = mysql_query($sql);
if ($res) $rs = mysql_fetch_array($res);

if ($session_userid=="" || $rs[userid]==="") {
		$session_userid="";
		echo "
		<script>
		location.replace('login_form.php');
		</script>
		";
		die;
}
?>
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<link href="global.css" rel="stylesheet" type="text/css">
```

```
<script language="JavaScript">
<!--
function na_open_window(name, url, left, top, width, height, toolbar, menubar, statusbar, scrollbar,
resizable)
{
   toolbar_str = toolbar ? 'yes' : 'no';
   menubar_str = menubar ? 'yes' : 'no';
   statusbar_str = statusbar ? 'yes' : 'no';
   scrollbar_str = scrollbar ? 'yes' : 'no';
   resizable_str = resizable ? 'yes' : 'no';

   cookie_str = document.cookie;
   cookie_str.toString();

   pos_start   = cookie_str.indexOf(name);
   pos_end     = cookie_str.indexOf('=', pos_start);

   cookie_name = cookie_str.substring(pos_start, pos_end);

   pos_start   = cookie_str.indexOf(name);
   pos_start   = cookie_str.indexOf('=', pos_start);
   pos_end     = cookie_str.indexOf(';', pos_start);

   if (pos_end <= 0) pos_end = cookie_str.length;
   cookie_val = cookie_str.substring(pos_start + 1, pos_end);
   if (cookie_name == name && cookie_val  == "done")
      return;

window.open(url,name,'left='+left+',top='+top+',width='+width+',height='+height+',tool
bar='+toolbar_str+',menubar='+menubar_str+',status='+statusbar_str+',scrollbars='+sc
rollbar_str+',resizable='+resizable_str);
}

// -->
</script>

</head>
```

```
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<form name="login_form" id="login_form" style="margin:0px;" target="_blank">
    <p> </p>
    <table border="0" width="835" align="center" cellpadding="0" cellspacing="0" height="821">
        <tr>
            <td width="835" height="11" colspan="2"><iframe name="menu" src="../mainmenurr.htm" width="832"
height="140" vspace="0" hspace="0" marginwidth="0" marginheight="0" frameborder="0"></iframe></td>
        </tr>
        <tr>
            <td width="407" height="86">
                <p><b>&quot;<? echo $rs[name];?>[<? echo $rs[userid];?>]&quot;</b>님
<br>
        <br>
        환영합니다.</p>
                <p><input type="button" name="Button" value="로그아웃"
 onClick="location.replace('logout.php');
"style="background-color:white;"><input type="button" name="Button" value="회원정보수정"
onClick="main.location='../user/edit_form.php';na_open_window('회원정보수정','../user/edit_form.php',
100, 100, 300, 200, 0, 0, 0, 0, 0);" style="background-color:white;"></p>
             </td>
            <td width="428" height="281" rowspan="4"><img src="mainimage.jpg" width="422" height="285"
border="0"></td>
        </tr>
        <tr>
            <td width="407" height="19">
                <p><input type="button" name="Button" value="암호변경"
onClick="main.location='../user/change_password_form.php';"
style="background-color:white;"><input type="button" name="Button" value="탈퇴"
onClick="if (confirm('정말로 탈퇴하시겠습니까?\r\n\r\n탈퇴하시면 개인정보보호정책에 따라
회원정보가 완전히 삭제됩니다.')) location.replace('../user/resign.php');" style="background-color:white;"></p>
            </td>
        </tr>
        <tr>
            <td width="407" height="24">
                <p align="left"> </p>
            </td>
        </tr>
        <tr>
            <td width="407" height="124">
```

```
                <p> </p>
                <p> </p>
                <p><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,29,0" width="347" height="29">
                <param name="movie" value="clock.swf">
                <param name="play" value="true">
                <param name="loop" value="true">
                <param name="quality" value="high">
<embed width="347" height="29" src="clock.swf" play="true" loop="true" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Versi
on=ShockwaveFlash"></object></p>
            </td>
        </tr>
        <tr>
            <td width="835" height="15" colspan="2"><iframe name="main"
 src="../main.php" width="847" height="350" vspace="0" hspace="0" marginwidth="0"
 marginheight="0" frameborder="0"></iframe></td>
        </tr>
        <tr>
            <td width="835" height="11" colspan="2"><img src="copyright.jpg" width="832"
height="50" border="0"></td>
        </tr>
    </table>
    <p> </p>
</form>
</body>
</html>
```

(9) resign.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

if ($session_userid==="") {
        echo "
        <script>
        alert('[회원전용] 로그인 후 사용하십시오.');
```

```
		location.replace('../login/login_form.php');
		</script>
		";
		die;
}

//데이터베이스에 등록합니다.
$sql = "delete from user where userid='$session_userid'";
$res = mysql_query($sql);
$affected_rows = mysql_affected_rows();
if ($affected_rows>0) {
		$session_userid="";
		echo "
		<script>
		alert('[탈퇴성공] 정상적으로 회원에서 탈퇴하셨습니다.');
		location.replace('../login/login_form.php');
		</script>
		";
} else {
		echo "
		<script>
		alert('[탈퇴실패] 데이터베이스서버의 오류 또는 회원필드 오류로 인하여 탈퇴실패하였습니다.');
		location.back();
		</script>
		";
}
?>
```

3.1.7. login 폴더

(1) login_form.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

if ($session_userid!="") {
        echo "
        <script>
        location.repl ace('../login/login_success.php');
        </script>
        ";
        die;
}
?>
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<link href="global.css" rel="stylesheet" type="text/css">

<script language="JavaScript">
<!--
function na_open_window(name, url, left, top, width, height, toolbar, menubar, statusbar,
scrollbar, resizable)
{
  toolbar_str = toolbar ? 'yes' : 'no';
  menubar_str = menubar ? 'yes' : 'no';
  statusbar_str = statusbar ? 'yes' : 'no';
  scrollbar_str = scrollbar ? 'yes' : 'no';
  resizable_str = resizable ? 'yes' : 'no';

  cookie_str = document.cookie;
  cookie_str.toString();

  pos_start  = cookie_str.indexOf(name);
```

```
  pos_end    = cookie_str.indexOf('=', pos_start);

  cookie_name = cookie_str.substring(pos_start, pos_end);

  pos_start  = cookie_str.indexOf(name);
  pos_start  = cookie_str.indexOf('=', pos_start);
  pos_end    = cookie_str.indexOf(';', pos_start);

  if (pos_end <= 0) pos_end = cookie_str.length;
  cookie_val = cookie_str.substring(pos_start + 1, pos_end);
  if (cookie_name == name && cookie_val  == "done")
    return;

  window.open(url, name,'left='+left+',top='+top+',width='+width+',height='+height+',
toolbar='+toolbar_str+',menubar='+menubar_str+',status='+statusbar_str+',scrollbars='+scrollbar_str+'
,resizable='+resizable_str);
}

// -->
</script>

</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<br>
<br>
<form action="../login/login_engine.php" method="post" name="login_form" id="login_form"
style="margin:0px;" onSubmit="return checkForm(this.form);">
  <table width="705" border="0" align="center" cellpadding="0" cellspacing="0" bordercolorlight="white"
height="641">
        <tr>
            <td width="705" align="right" bgcolor="white" colspan="3">
                <p align="left"><img src="images/loginup.jpg" width="700" height="185"
border="0">  </p>
            </td>
        </tr>
        <tr>
            <td width="342" height="333" align="right" bgcolor="white" rowspan="5"><img
src="images/loginleft.jpg" width="306" height="330" border="0"></td>
```

```
            <td width="120" height="33" bgcolor="white" bordercolorlight="white"><img
src="images/ID.jpg" width="110" height="30" border="0">                </td>
            <td width="243" height="33" bgcolor="white" bordercolorlight="white"><input name="userid"
type="text" id="userid" size="30" style="border:1px #333333 solid;width:160px;height:25px;" maxlength="10">
            </td>
        </tr>
        <tr>
            <td width="120" height="39" bgcolor="white" bordercolorlight="white"><img
src="images/PW.jpg" width="110" height="30" border="0"></td>
            <td width="243" height="39" bgcolor="white" bordercolorlight="white"><input
name="password" type="password" id="password" size="30" style="border:1px #333333
solid;width:160px;height:25px;" maxlength="10">
            </td>
        </tr>
        <tr>
            <td width="120" height="39" bgcolor="white" bordercolorlight="white"><img
src="images/otp.jpg" width="110" height="30" border="0"></td>
            <td width="243" height="39" bgcolor="white" bordercolorlight="white"><input name="r"
type="password" id="UniKey" size="30" style="border:1px #333333 solid;width:160px;height:25px;"
maxlength="10">
            </td>
        </tr>
        <tr>
            <td width="129" height="113" bgcolor="white" bordercolorlight="white">
                <p align="center"><img src="images/leftarrow.jpg" width="82" height="37"
border="0"></p>
            </td>
            <td width="265" height="113" bgcolor="white" bordercolorlight="white">
                <p><b><u><font face="휴먼둥근헤드라인" color="red"><span style="font-size:32pt;
background-color:white;">
<?php
mt_srand((double)microtime()*1000000);
$c = rand(10000, 99999);
echo $c
?> //의사난수 랜덤값 추출
</span></font></u></b><font size="7" color="red"><input type="hidden" name="cc" value="<?php
echo $c ?>" size="12">
</font></p>
                <p>     <input type="submit" name="Submit"
value="인증" style="background-color:white; border-color:white;">
```

```
<input type="reset" name="Reset" value="취소" style="background-color:white; border-color:white;">
</p>
                </td>
            </tr>
            <tr>
                <td width="363" height="88" colspan="2" bgcolor="white" bordercolorlight="white"><img
src="images/login06.jpg" width="394" height="82" border="0"></td>
            </tr>
            <tr>
                <td width="705" height="105" align="right" bgcolor="white" colspan="3">
                    <p align="left">
<img src="images/logindown.jpg" width="700" height="85" border="0" usemap="#ImageMap1"></p>
                </td>
            </tr>
    </table>
    <br>
      <p> </p>
</form>
<script>
function checkForm(theForm) {
            if (theForm.userid.value==="") {
                        alert("아이디를 입력하십시오.");
                        theForm.userid.focus();
                        return false;
            } else {
                        return true;
            }
}
</script>
<map name="ImageMap1">
<area shape="rect" coords="76, 11, 193, 35" href="javascript:na_open_window('회원가입
', 'http://61.81.108.24/user/add_form.php', 0, 0, 510, 470, 0, 0, 0, 0, 0)" target="_self">
</map></body>
</html>
```

(2) login_engine.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

$sql = "select * from user where userid='$userid'";
$res = mysql_query($sql);
if ($res) $rs = mysql_fetch_array($res);

    $r = (int)$_POST['r'];
        $cc = (int)$_POST['cc'];
        $master_key = 'abcd';
        $c = $master_key . $cc;
        $hash = sha1($c);
        $h0 = ord($hash{0});
        $h1 = ord($hash{1});
        $h1 = $h1 * 256;
        $h = (int)$h0 + (int)$h1;
        if ($h < 0) $h = $h * -1;

if ($rs[userid]==$userid && $userid!="") {//아이디 일치
        if ($rs[password]==md5($password)) {//암호 일치

                if ($r == $h) { // 고유키 일치
                $session_userid = $rs[userid];//인증완료
                echo "
                <script>
                location.replace('login_success.php');
                </script>
                ";
        } else {//고유키 오류
                echo "
                <script>
                alert('[인증실패] 고유키가 올바르지 않습니다.');
                location.replace('login_form.php');
```

```
                </script>
                ";
        }

        } else {//암호오류
                echo "
                <script>
                alert('[인증실패] 암호가 올바르지 않습니다.');
                location.replace('login_form.php');
                </script>
                ";
        }
} else {//아이디 오류
        echo "
                <script>
                alert('[인증실패] 아이디가 올바르지 않습니다.');
                location.replace('login_form.php');
                </script>
                ";
}
?>
```

(3) login_sucess.php

```
<?
include_once("../3Teamlib/session.php");
include_once("../3Teamlib/dbcon.php");

$sql = "select * from user where userid='$session_userid'";
$res = mysql_query($sql);
if ($res) $rs = mysql_fetch_array($res);

if ($session_userid=="" || $rs[userid]==="") {
        $session_userid==="";
        echo "
        <script>
```

```
            location.replace('login_form.php');
            </script>
            ";
            die;
}
?>
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<link href="global.css" rel="stylesheet" type="text/css">



<script language="JavaScript">
<!--
function na_open_window(name, url, left, top, width, height, toolbar, menubar, statusbar, scrollbar, resizable)
{
   toolbar_str = toolbar ? 'yes' : 'no';
   menubar_str = menubar ? 'yes' : 'no';
   statusbar_str = statusbar ? 'yes' : 'no';
   scrollbar_str = scrollbar ? 'yes' : 'no';
   resizable_str = resizable ? 'yes' : 'no';

   cookie_str = document.cookie;
   cookie_str.toString();

   pos_start   = cookie_str.indexOf(name);
   pos_end     = cookie_str.indexOf('=', pos_start);

   cookie_name = cookie_str.substring(pos_start, pos_end);

   pos_start   = cookie_str.indexOf(name);
   pos_start   = cookie_str.indexOf('=', pos_start);
   pos_end     = cookie_str.indexOf(';', pos_start);

   if (pos_end <= 0) pos_end = cookie_str.length;
   cookie_val = cookie_str.substring(pos_start + 1, pos_end);
   if (cookie_name == name && cookie_val  == "done")
```

```
    return;

  window.open(url, name, 'left='+left+',top='+top+',width='+width+',height='+height+',
toolbar='+toolbar_str+',menubar='+menubar_str+',status='+statusbar_str+',scrollbars='+scrollbar
_str+',resizable='+resizable_str);
}

// -->
</script>

</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<form name="login_form" id="login_form" style="margin:0px;">
    <p> </p>
    <table width="835" align="center" cellpadding="0" cellspacing="0" height="815"
style="border-collapse:collapse;">
        <tr>
            <td width="835" height="11" colspan="2" style="border-width:1; border-color:rgb(153,153,153);
border-style:none;"><iframe name="menu" src="../mainmenurr.htm" width="832" height="140" vspace="0"
hspace="0" marginwidth="0" marginheight="0" frameborder="0"></iframe></td>
        </tr>
        <tr>
            <td width="407" height="83" style="border-width:1; border-color:rgb(153,153,153);
border-style:none;">

<p><b>           &n
bsp;      &quot;<? echo $rs[name];?>[<? echo
$rs[userid];?>]&quot;</b>님
<br>

            &nbs
p;           &n
bsp;환영합니다.</p>
                <p>    <input type="button" name="Button"
value="로그아웃" onClick="location.replace('logout.php');"
style="background-color:white;"><input type="button" name="Button" value="암호변경"
style="background-color:white;" OnClick="na_open_window('암호변경', '../user/change
_password_form.php', 100, 100, 450, 300, 0, 0, 0, 0, 0);"><input type="button"name=
```

```
"Button" value="회원정보수정" style="background-color:white;"OnClick="na_open_
window('회원정보수정', '../user/edit_form.php',
100, 100, 500, 400, 0, 0, 0, 0, 0);"><input type="button" name="Button" value="탈퇴"
onClick="if (confirm('정말로 탈퇴하시겠습니까?\r\n\r\n탈퇴하시면 개인정보보호정책에 따라
회원정보가 완전히 삭제됩니다.'))
 location.replace('../user/resign.php');" style="background-color:white;"></p>
            </td>
            <td width="428" height="275" rowspan="4" style="border-width:1; border-color:rgb(153,153,153);
border-style:none;"><img src="mainimage.jpg" width="422" height="285" border="0"></td>
        </tr>
        <tr>
            <td width="407" height="19" style="border-width:1; border-color:rgb(153,153,153);
border-style:none;">
                <p> </p>
            </td>
        </tr>
        <tr>
            <td width="407" height="24" style="border-width:1; border-color:rgb(153,153,153);
border-style:none;">
                <p align="left"> </p>
            </td>
        </tr>
        <tr>
            <td width="407" height="124" style="border-width:1; border-color:rgb(153,153,153);
border-style:none;">
                <p> </p>
                <p> </p>
                <p><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#vers
ion=6,0,29,0" width="347" height="29">
                <param name="movie" value="clock.swf">
                <param name="play" value="true">
                <param name="loop" value="true">
                <param name="quality" value="high">
<embed width="347" height="29" src="clock.swf" play="true" loop="true" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Versi
on=ShockwaveFlash"></object></p>
            </td>
        </tr>
```

```
        <tr>
            <td width="835" height="15" colspan="2" style="border-width:1; border-color:rgb(153,153,153); border-style:none;"><iframe name="main" src="../main.php" width="847" height="350" vspace="0" hspace="0" marginwidth="0" marginheight="0" frameborder="0"></iframe></td>
        </tr>
        <tr>
            <td width="835" height="11" colspan="2" style="border-width:1; border-color:rgb(153,153,153); border-style:none;"><img src="copyright.jpg" width="832" height="50" border="0"></td>
        </tr>
    </table>
    <p> </p>
</form>
</body>
</html>
```

(3) logout.php

```
<?
include_once("../3Teamlib/session.php");

$session_userid = "";//인증해제
echo "
<script>
location.replace('login_form.php');
</script>
";
?>
```

3.1.8. 3Teamlib 폴더

(1) dbcon.php

```
<?
/////데이터베이스 연결계정 정의 시작///////
$db_hostname="localhost";
$db_username="root";
$db_password="3wh3wh";
$dbname = "3team";
/////데이터베이스 연결계정 정의 끝///////
$sock = mysql_connect($db_hostname,$db_username,$db_password)
                or die ("데이터베이스 서버에 연결할 수 없습니다.");
$db = mysql_select_db($dbname,$sock)
                or die ("'$dbname' 데이터베이스를 찾을 수 없거나 접근권한이 없습니다.");
?>
```

(2) function.php

```
<?
function str_limit($string,$limit_length,$add_string){
        $full_length=strlen($string);
        for($k=0; $k<$limit_length-1; $k++){
                if(ord(substr($string, $k, 1))>127) $k++;
        }
        if ($full_length > $limit_length){
                $final_string=substr($string, 0, $k).$add_string;
        } else {
                $final_string=$string;
        }
        return $final_string;
}
?>
```

(3) session.php

```
<?
session_start();
$session_userid=$HTTP_SESSION_VARS["session_userid"];
if (!session_is_registered ("session_userid")) session_register("session_userid");
$session_auth=$HTTP_SESSION_VARS["session_auth"];
if (!session_is_registered ("session_auth")) session_register("session_auth");
?>
```

3.1.9. 작동법

<그림15> 인터넷 사용자 인증번호와 스마트폰 OTP생성 화면

3조 어플리케이션 파일(mobileotp.apk) 실제 스마트폰 적용화면

<그림16> 3조 어플리케이션(mobileotp.apk 파일)설치중

http://isotp.joongbu.ac.kr/mobileotp.apk 혹은 http://61.81.108.24/mobileotp.apk 으로 접속하면 자동으로 다운 및 설치가 가능하다.(안드로이드 스마트폰 기준)

<그림17> 안드로이드 스마트폰 메인화면 UI에 등록된 3Team O.T.P 어플리케이션 아이콘

4. 결론

지식정보가 가장 큰 자산이 되는 시대에 따라 보안 규모 또한 커지고 있지만 개인 정보 유출의 심각성은 이제 어린 아이들도 안다고 할 만큼 빈번히 일어나고 있을 뿐더러 pc방, 직장, 학교 등 여러 곳에서 ID, PW를 쓰는 일이 많은 만큼 내 비밀 번호의 노출 빈도 또한 높다.

알툴즈 사업본부 에서는 pc 사용자가 입력한 ID/PW 정보를 가로채 일본으로 유출시키는 **키로거(Keylogger)**형 악성코드가 인터넷 카페와 블로그의 첨부 파일을 통해 확산 중이며, 알약 고객지원팀을 통해 감염 피해 신고 건수가 늘고 있다고 발표했다. 위험은 먼 곳에 있는 것이 아니며, 이제는 단지 친구의 이야기가 아니라는 것이다. 자신도 모르는 사이 노출된 PW를 통해 나의 정보가 여기 저기 사용되고 또 그것이 해외로까지 넘어갔을 때의 그 피해는 상상할 수도 없는 일이다. 가장 좋은 방법은 이러한 일이 일어나기 전에 최대한 예방하는 것이다. 그에 따라서 가장 화두가 되고 있는 OTP는 사용자의 안전을 위한 방편으로 최선이 될 것이다.

이에 본 연구에서는 OTP의 기본구조를 살피고 그중에서도 편의성을 가진 **안드로이드 모바일 O.T.P 어플리케이션과** 기본 작동 시스템 만들어 실질적인 OTP의 발급과 사용까지 구현해보았다.

앞으로 OTP 시장의 규모는 엄청나게 커질 것이며 OTP가 가져올 기대효과 또한 엄청날 수 있다. OTP는 이제 생소하기 보다는 바로 실생활에서 흔히 사용하는 기술이 될 것이다. 하지만 기존의 체계에 반한 부작용 또한 만만치 않을 것이며 아래와 같은 사항들을 고려하여 OTP 도입의 문제를 최소화 시키고 기존 체제보다 큰 보안 효과를 이끌어내야 할 것이다.