Snort를 이용한 IDS구축

2005. 5.

인프라보호단 / 보안관리팀

KISA 한국정보보호진흥원
Korea Information Security Agency

목 차

Ⅰ. 서 론.....1

Ⅱ. Snort를 이용한 IDS 구축.....1
1. Snort란 무엇인가?.....1
2. Snort의 구조.....2
3. Snort설치 및 설정.....3

<별첨 #1> snort 로그의 다양한 저장법.....21
<별첨 #2> snort 실행 모드.....23
<별첨 #3> snort 룰 이해하기.....27
<별첨 #4> snort에 관련된 유용한 프로그램 소개.....29

그림 차례

[그림 1] snort 홈페이지.....2
[그림 2] libpcap 다운로드 및 압축해제.....4
[그림 3] libpcap 컴파일.....4
[그림 4] snort 다운로드 및 압축해제.....4
[그림 5] snort 컴파일.....5
[그림 6] snort.conf의 기본 설정.....6
[그림 7] snort.conf의 기본 설정..... 7
[그림 8] snort.conf의 기본 설정.....8
[그림 9] snort.conf의 기본 설정.....8
[그림 10] snort.conf의 기본 설정.....9
[그림 11] snort.conf의 기본 설정.....10
[그림 12] snort.conf의 기본 설정.....10

[그림 13] snort.conf의 기본 설정.....11
[그림 14] snort의 기본 설정.....11
[그림 15] snort.conf의 기본 설정.....12
[그림 16] snort의 기본 설정.....12
[그림 17] snort 기본 실행 화면.....13
[그림 18] snort 실행.....13
[그림 19] snort 홈페이지의 룰 다운로드 회원 가입.....15
[그림 20] oinkmaster 홈페이지.....16
[그림 21] oinkmaster 다운로드 및 압축해제.....16
[그림 22] snort 홈페이지 가입 후 수신한 메일.....17
[그림 23] onikmaster 다운로드 코드 획득.....17
[그림 24] oinkmaster.conf의 기본 설정.....18
[그림 25] oinkmaster.conf의 기본 설정.....18
[그림 26] oinkmaster.conf의 기본 설정.....19
[그림 27] oinkmaster.conf의 기본 설정.....20
[그림 28] Message 로 SID 찾는 URL.....20
[그림 29] SCAN 으로 질의한 결과.....21
[그림 30] 특정 sid 에 대한 자세한 설명.....21
[그림 31] oinkmaster 실행 화면.....22
[그림 32] oinkmaster 실행 화면.....22
[그림 33] snort 실행으로 생성된 로그.....24
[그림 34] log 파일로 저장하는 설정.....28
[그림 35] 생성된 로그 파일.....28
[그림 36] snort.conf 에러시 화면.....29
[그림 37] snort.conf 에러시 화면.....30
[그림 38] mysql 과 연동하여 snort 설치.....34
[그림 39] acid 의 메인화면.....37
[그림 40] acid 의 이벤트 질의결과.....37
[그림 41] 탐지한 패킷의 디코딩 화면.....38

Ⅰ. 서론

방화벽(firewall)과 함께 또 하나의 주목받는 보안 솔루션으로는 침입탐지시스템이라 불리는 IDS (Intrusion Detection System)가 있다. 앞에서 살펴보았던 방화벽이 IP나 포트를 기준으로 비정상 트래픽을 차단하는 것이라면 IDS는 포트에 대한 정보뿐만 아니라 패킷의 데이터까지 분석하여 정상적인 트래픽 여부를 결정하는 것으로 이를 통해 실제로 인터넷을 통해 어떠한 위협이 발생하고 있는지를 분석할 수 있게 된다. 이를테면 사내 게이트웨이에 IDS를 설치해 둔다면 사내를 오가는 트래픽을 분석하여 비정상적인 트래픽이 보인다면 바로 로그에 남기도록 할 수 있다.
그리고, 방화벽의 경우 열려있는 포트인 80번을 통해 worm 등과 같은 비정상적인 공격이 있을 경우에는 차단할 수 있는 방법이 없으나, IDS 에서는 이러한 공격을 인식할 수 있는 기능이 있다. 따라서 IDS와 방화벽을 연동하여 IDS 에 탐지한 비정상 트래픽을 차단하는 기능을 갖춘 솔루션도 있다. 그러나 그렇다고 해서 IDS 가 모든 공격을 100% 정확하게 인식하여 탐지하는 것은 아니며, 몇 가지 근본적인 한계를 가지고 있다. 일단 가장 큰 문제는 “오탐율”이다. IDS에서 오탐은 크게 두 부류로 나눌 수 있는데, 실제로 비정상 트래픽임에도 불구하고 이를 탐지하지 못하는 경우를 “False Negative”라 하며, 비정상 트래픽이 아닌 정상적인 트래픽을 탐지하는 경우를 “False Positive"라 한다. 실제 IDS를 운영하다 보면 위와 같은 오탐이 매우 높다는 것을 알 수 있으며 이 오탐율을 줄이는 것이 IDS의 관건이라 할 수 있겠다. 또한 앞에서 언급한 바와 같이 침입에 대한 탐지는 할 수 있어도 이 공격에 대한 대응은 부족하다는 한계가 있다.

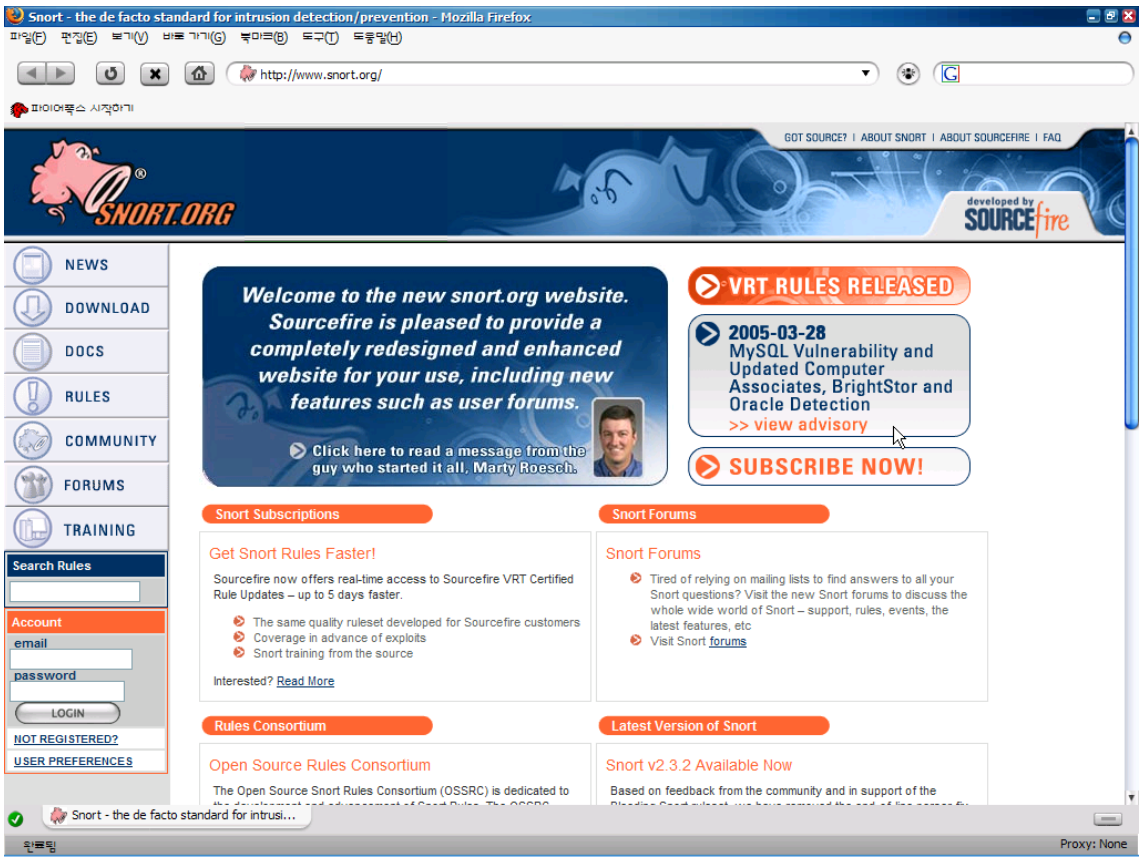
Ⅱ. Snort를 이용한 IDS 구축

IDS가 침입을 탐지하는 방식은 몇 가지가 있지만 가장 대표적인 방식은 룰 기반의 패턴 매칭 방식으로 이는 사전에 정의된 패턴 또는 룰에 따라 트래픽이 매칭되었을 경우 공격 또는 비정상 트래픽으로 판단하는 것으로 이를테면, 80번 http포트를 통해 “cmd.exe”라는 URI 요청이 보일 경우 비정상 트래픽이라 판단하는 것이다. 여기에서는 이러한 룰 기반의 공개 IDS프로그램 중 가장 대중적으로 사용되고 있는 snort 라는 프로그램을 이용하여 비정상 트래픽을 탐지할 수 있는 방안에 대해 알아보도록 하겠다.

1. Snort 란 무엇인가?

Snort는 “sniffer and more"라는 말에서 유래되었는데, 처음 공개되었을 때는 코드도 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. snort는 공식 홈페이지인 http://www.snort.org를 통해 지속적인 업그레이드를 제공하고 있다.



[그림 1] snort 홈페이지

snort의 기능은 간단히 다음과 같이 분류할 수 있다.

패킷 스니퍼(sniffer)	네트워크의 패킷을 읽어 보여주는 기능
패킷 로거(logger)	모니터링 한 패킷을 저장하고 로그에 남기는 기능
network IDS	네트워크 트래픽을 분석하여 공격을 탐지하는 기능으로 buffer overflow나 port scan, IP scan 등 대부분의 공격을 탐지할 수 있다.

snort는 오픈 소스로 개발중인 패킷 캡처 라이브러리인 libpcap을 사용하여 패킷을 캡처하고, 수집된 패킷이 사전에 정의된 snort 공격 룰과 비교하여 만약 매칭 되었을 경우 syslog를 통해 로그를 남기거나 특정 디렉토리의 특정 파일 또는 database에 남기도록 할 수 있다.

2. snort 의 구조

snort프로그램은 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.

① 스니퍼(sniffer)
② preprocessor
③ 탐지엔진
④ 로깅(출력)

snort는 먼저 스니퍼를 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 인코딩 플러그인이나 포트스캔 등 몇 가지 플러그인을 먼저 거치면서 매칭이 되는지 확인하게 된다. 물론 preprocessor 역시 모듈화 되어 있어 각자의 환경에 불필요하다면 disable 할 수 있다. 이를테면 RPC 트래픽에 대해 탐지할 필요가 없다면 RPC 관련 preprocessor를 주석처리하면 된다.
그리고 preprocessor를 통과한 패킷은 snort IDS의 핵심이라 할 수 있는 룰 기반의 탐지엔진을 거치면서 사전에 정의된 탐지룰과 매칭되는지 확인하게 된다. 만약 룰에 매칭되었을 경우에는 사전에 정의된 정책에 따라 로그에 남게 되고, 그렇지 않은 경우 통과를 하게 된다.

3. snort 설치 및 설정

snort가 패킷을 스니핑 하려면 패킷캡처 라이브러리인 libpcap을 기반으로 동작하므로 snort를 설치하기 전에 libpcap이 먼저 설치되어 있어야 한다. 설치는 다음과 같은 순서를 따라서 하면 된다.

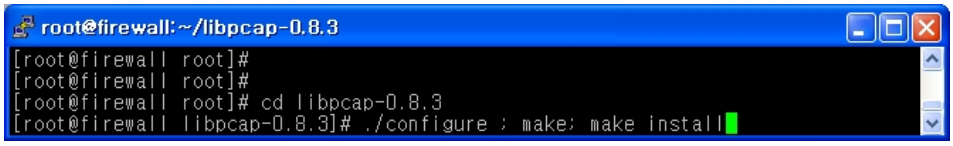
STEP 1. libpcap 다운로드 및 설치

wget http://www.tcpdump.org/release/libpcap-0.8.3.tar.gz (파일 다운로드)
tar zxvfp libpcap-0.8.3.tar.gz (압축 풀기)

[그림 2] libpcap 다운로드 및 압축해제

libpcap은 http://www.tcpdump.org/에서 최신 버전을 다운로드하여 설치하면 된다.

cd libpcap-0.8.3 (압축 푼 파일 디렉토리 이동)
./configure ; make ; make install (컴파일 및 설치)

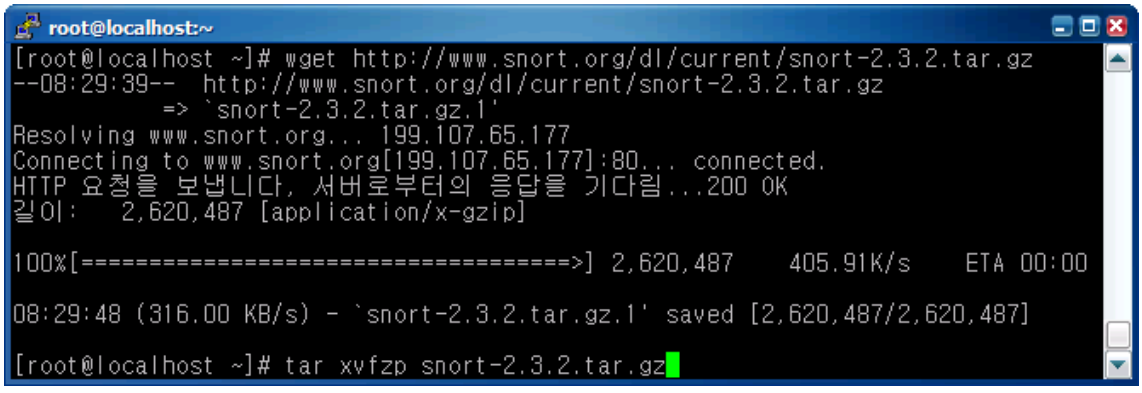


[그림 3] libpcap 컴파일

만약 설치가 잘 안될 경우에는 http://www.rpmfind.net/에서 각자 배포판 버전에 맞는 rpm 파일을 다운로드하여 설치해도 된다.

STEP 2. snort 다운로드 및 설치

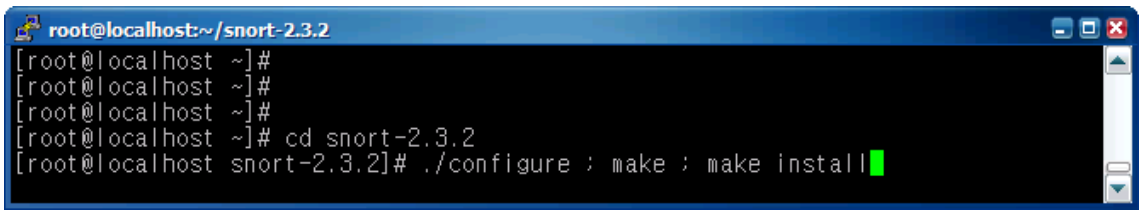
wget http://www.snort.org/dl/current/snort-2.3.2.tar.gz (snort 파일 다운로드)
tar xvfzp snort-2.3.2.tar.gz (압축 풀기)



[그림 4] snort 다운로드 및 압축해제

libpcap을 설치한 후 snort를 설치하도록 한다. snort는 http://www.snort.org/dl/

에서 최신 버전의 소스파일을 다운로드 후 압축 해제한다.
cd snort-2.3.2 (압축 푼 파일 디렉토리 이동)
./configure ; make ; make install (컴파일 및 설치)



[그림 5] snort 컴파일

같은 방법으로 configure로 Makefile을 만든 후 make; make install로 설치하면 된다.

위와 같이 컴파일하면 snort가 설치된 snort-2.3.x/src 디렉토리와 /usr/local/bin/ 디렉토리에 snort 라는 실행 파일이 생기고, snort-2.3.x/etc 디렉토리에는 snort.conf 등 설정 파일이 생성된다. 그리고 snort-2.2.x/rules 디렉토리에 침입 탐지를 위한 각종 룰 파일이 위치하게 된다. 만약 특정 디렉토리에 설치를 원하면 configure시 옵션을 주어 컴파일 할 수 있다.

STEP 3. snort.conf 설정

설치가 끝난 후에는 snort 설정 파일을 직접 설정해 보도록 한다. snort의 설정파일은 snort.conf로서 snort가 가동을 시작할 때 이 파일에 설정된 각종 변수 값 등을 읽어오게 된다. snort.conf는 크게 4부분으로 나뉘어져 있는데, 침입을 탐지할 ip 범위 등 network 관련 환경 변수를 지정하는 부분과, preprocessor를 지정하는 부분, 로깅 및 출력 등 output을 설정하는 부분과 마지막으로 침입을 탐지할 rule 파일등을 지정하는 부분이다.
별다른 설정 변경 없이 바로 사용해도 작동은 하지만 좀 더 정확하고 세부적인 정보를 위해 snort 설정파일의 변수 부분을 이해해야 할 필요가 있으므로, 아래의 내용을 숙지하여 자신의 환경에 맞게 수정하도록 한다.

① network 환경 설정

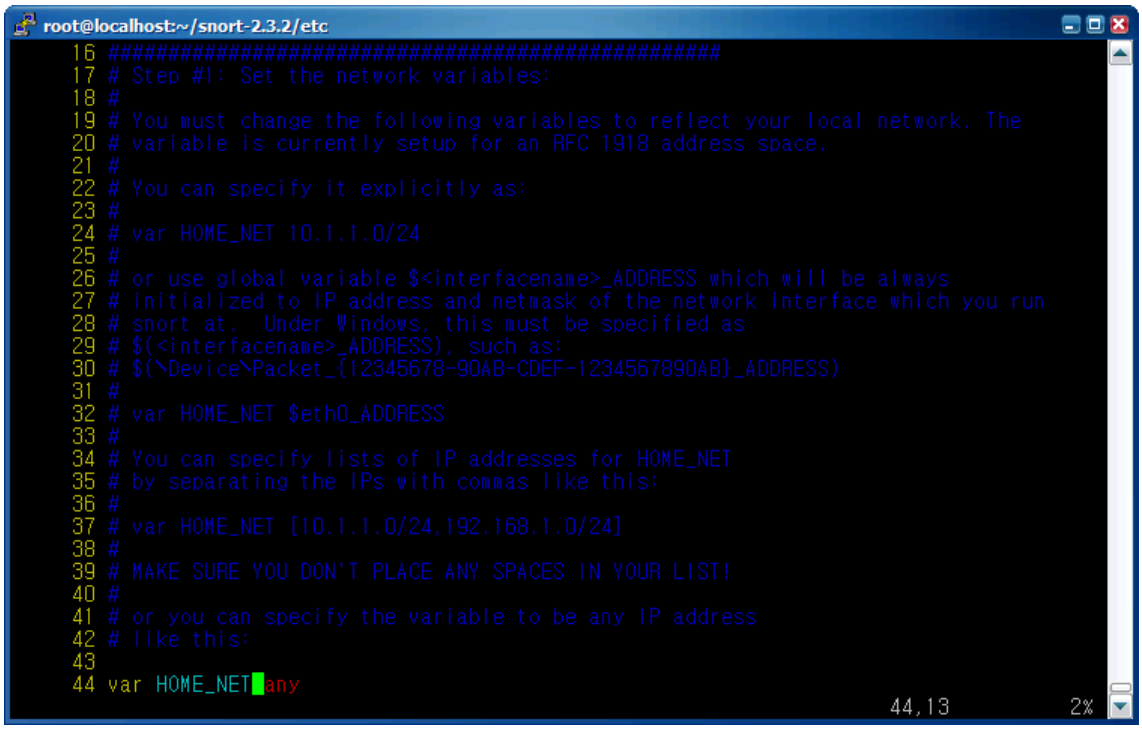
* var HOME_NET
  snort를 실행하여 트래픽을 탐지할 목적지 ip 또는 ip 대역을 지정하는 부분이다. snort에서 탐지되는 모든 목적지 ip에 대해 탐지하고자 한다면 '모든'의 의미인 any를 지정하면 되고 특정한 ip 또는 ip 대역만을 탐지하려면 해당 ip나 ip 대역을 지정하면 된다. 통상적으로 any로 지정하면 무난하다. 아래의 예제를 참조하도록 하자.

예제) var HOME_NET 10.1.1.0/24

: 10.1.1.0/24 또는 10.1.1.0/255.255.255.0 C class 대역을 목적지로 하는 패킷만을 감시하게 된다.

var HOME_NET [10.1.1.0/24,192.168.1.0/24]

: 10.1.1.0/24 C class 대역을 포함하여 192.168.1.0/24 C class 대역을 목적지로 하는 패킷도 감시하게 된다. ip 대역을 여러 곳 지정할 경우에는 위와 같이 공백 없이 콤마(,)를 지정하면 된다. 물론 특정한 ip 만을 감시하려면 해당 ip를 명시하면 된다.



[그림 6] snort.conf 의 기본 설정

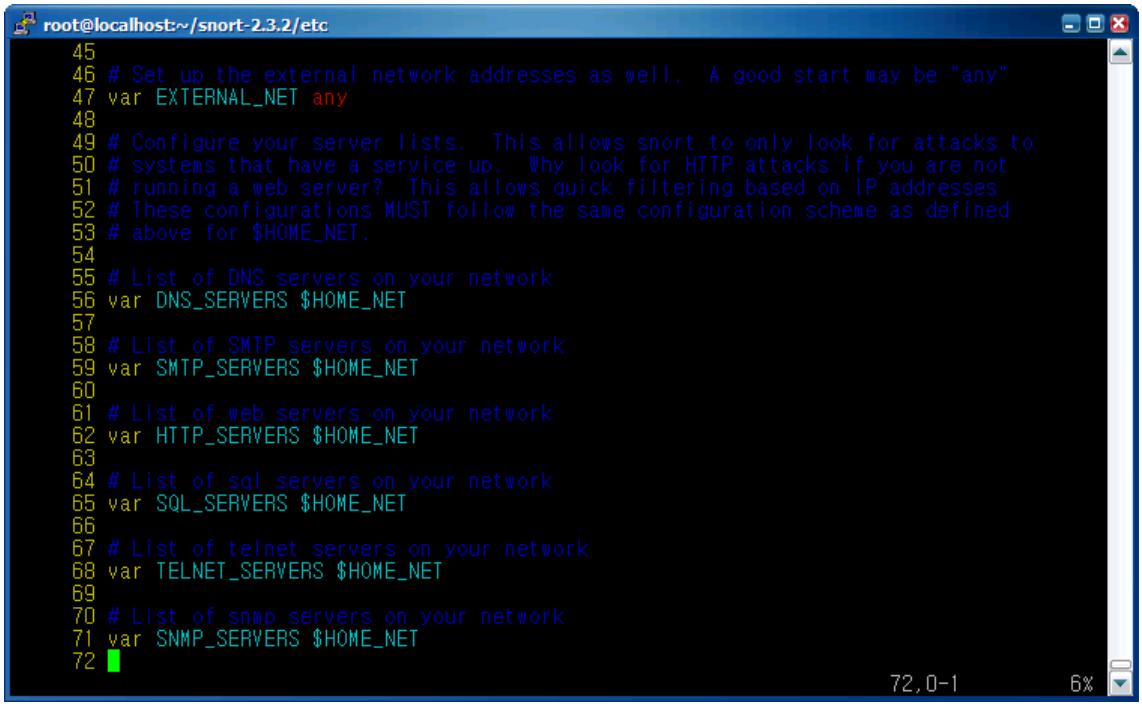
* EXTERNAL_NET any

HOME_NET이 침입을 탐지할 목적지 ip를 지정하는 부분이라면 EXTERNAL_NET은 이와 반대로 침입을 탐지할 패킷의 소스 ip 또는 ip 대역을 지정하는 부분이다. 만약 특정한 ip로부터 오는 패킷만을 감시하려면 해당 ip를 지정하면 된다. 통상적으로 any 로 지정하면 무난하다. 만약 특정한 ip 대역을 제외하고자 할 경우에는 아래와 같이 설정하면 된다.

var EXTERNAL_NET any ![192.168.1.3,10.1.0.0/24]

* var DNS_SERVERS $HOME_NET
* var SMTP_SERVERS $HOME_NET
* var HTTP_SERVERS $HOME_NET
* var SQL_SERVERS $HOME_NET
* var TELNET_SERVERS $HOME_NET
* var SNMP_SERVERS $HOME_NET

위는 서비스의 특성에 맞게 트래픽을 감시할 ip 또는 ip 대역을 지정하는 부분인데, 위의 예와 같이 $HOME_NET으로 지정하여 감시하도록 하는 것이 좋다.



[그림 7] snort.conf의 기본 설정

* var HTTP_PORTS 80

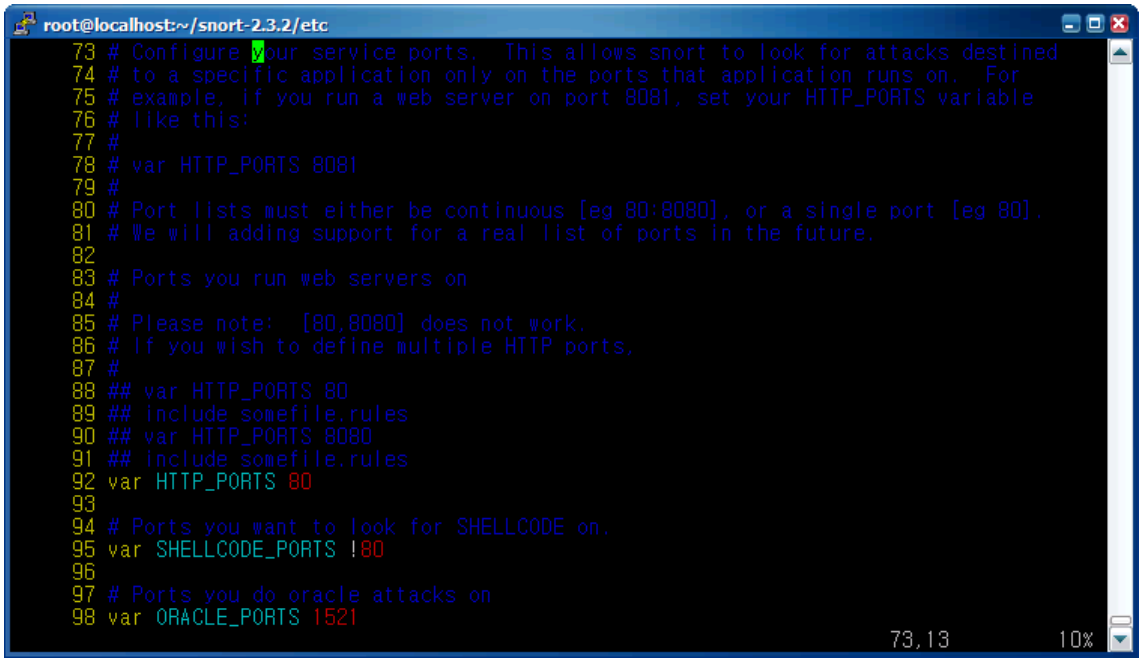
HTTP 트래픽을 모니터링하기 위하여 HTTP_PORTS를 몇 번으로 지정할 것인지 설정하면 된다. 일반적인 서비스 포트인 80번을 지정하면 된다.

* var SHELLCODE_PORTS !80

쉘코드(SHELLCODE)와 관련된 포트를 모니터링하기 위한 포트를 지정할 수 있는데, 기본값을 쓰도록 한다.

* var ORACLE_PORTS

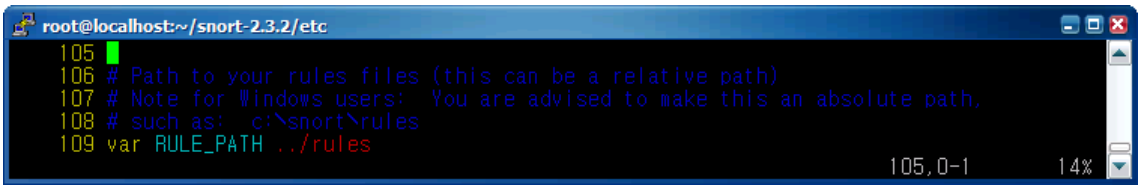
시스템에서 현재 구동되고 있는 오라클 서버의 포트를 지정하면 된다. 사용하지 않는다면 주석처리를 한다.



[그림 8] snort.conf 의 기본 설정

* var RULE_PATH ../rules

룰(rule) 파일이 있는 디렉토리를 지정한다. snort.conf 파일이 있는 경로에서의 상대경로로서 특별히 설정을 변경하지 않았다면 기본값을 쓰면 된다.



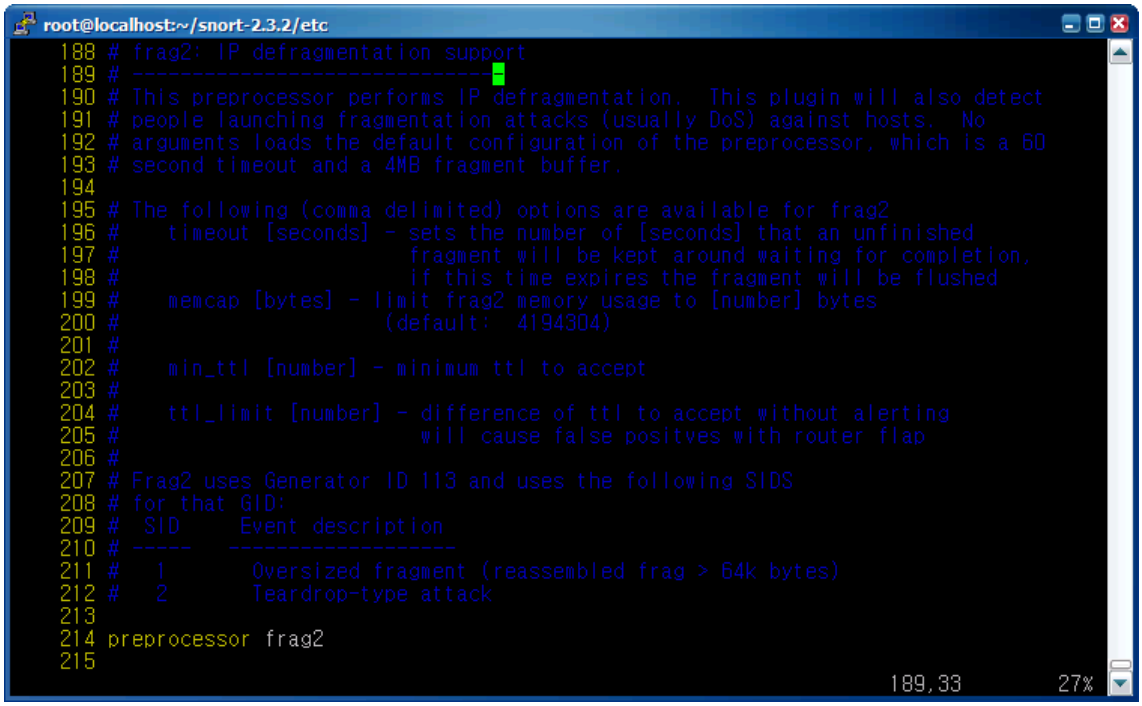
[그림 9] snort.conf의 기본 설정

② preprocessor 설정

preprocessor를 적절하게 사용하면 false positive나 false negative와 같은 오탐을 효과적으로 줄일 수 있다. 이 부분 역시 대부분은 기본값을 사용해도 무방하다.

* preprocessor frag2

frag2는 ip 단편화와 같은 서비스거부 공격을 탐지할 수 있다. 별도 설정 없이 기본값을 사용하면 된다.



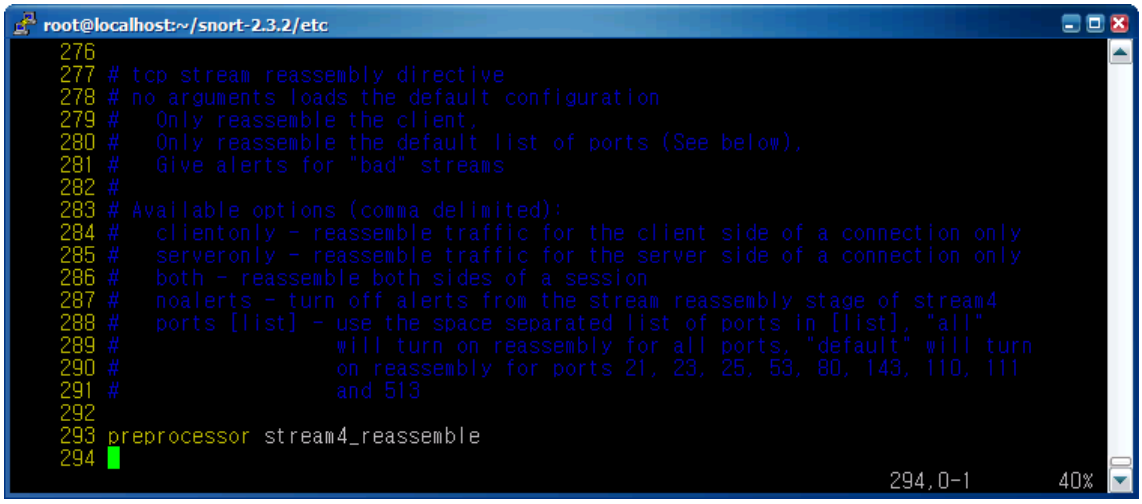
[그림 10] snort.conf의 기본 설정

* preprocessor stream4: detect_scans

  이 설정을 통해 다양한 스텔스(stealth) 포트 스캔이나 fingerprinting 등을 탐지할 수 있다. 여러 가지 옵션을 제공하는데, 기본 옵션을 사용해도 무방하다.

* preprocessor stream4_reassemble

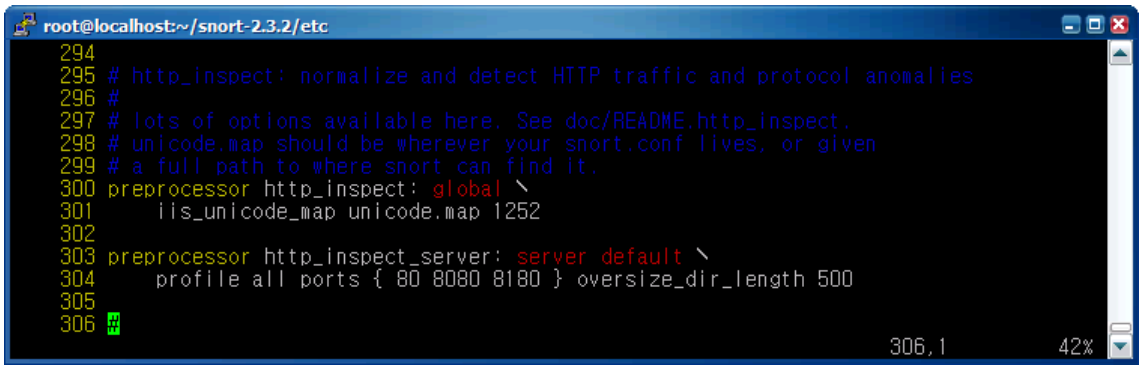
  stream4_reassemble는 바로 앞에서 언급한 stream4의 세션을 재조합하는 기능으로 clientonly, serveronly, both, noalerts, ports 의 5가지 옵션을 사용할 수 있는데, 아무런 값을 지정하지 않을 경우에는 기본적으로 clientonly와 기본포트(21, 23, 25, 53, 80, 143, 110, 111, 513)가 된다. clientonly는 클라이언트 측면에서의 트래픽 재조합을 하게 되고, serveronly는 서버 측면에서의 트래픽 재조합을 하게 되며 both를 지정하면 양 트래픽 모두에 대하여 재조합을 하게 된다. 그리고, ports는 어떤 포트와 관련된 트래픽에 대해 재조합을 할 것인지 지정하는 것인데, 꼭 필요한 정보만 설정하는 것이 좋다.



[그림 11] snort.conf의 기본 설정

* preprocessor http_inspect

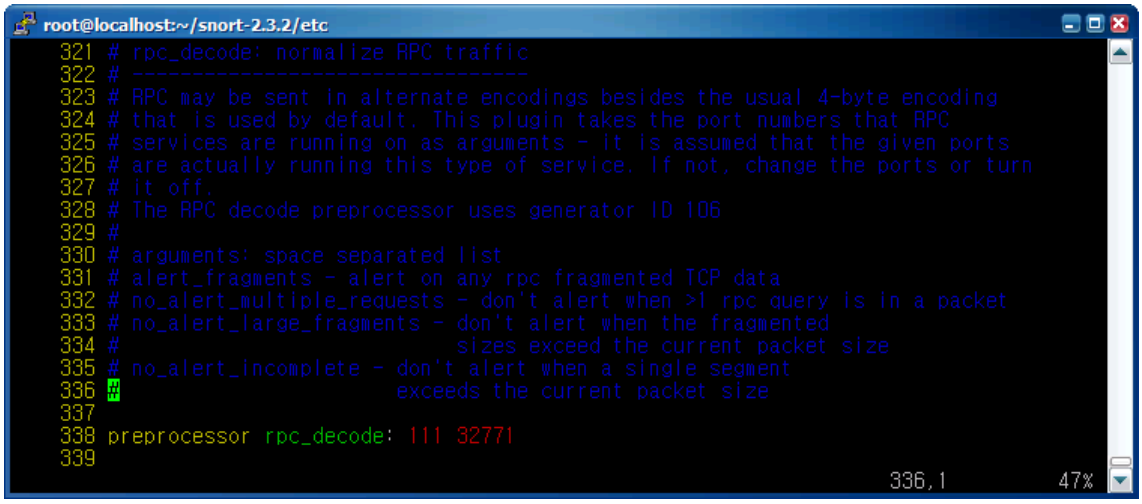
HTTP 트래픽을 탐지하기 위해 사용하는 설정으로 http port를 지정하고, 이를 통한 트래픽을 snort 기본 디렉토리의 etc/unicode.map에 저장된 룰셋을 바탕으로 탐지한다. 기본 설정으로 사용해도 무방하다.



[그림 12] snort.conf의 기본 설정

* preprocessor rpc_decode: 111 32771

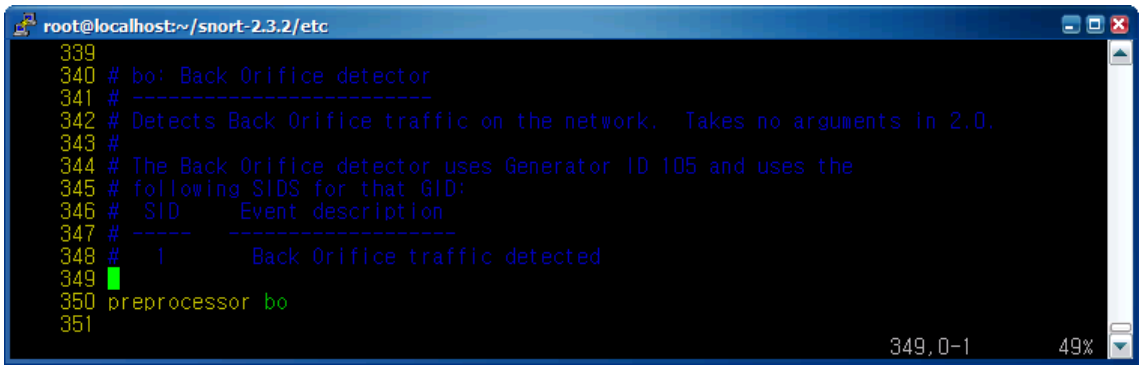
rpc_decode는 rpc에 기반한 단편화 공격을 탐지하기 위해 사용될 수 있는데, 통상적으로 rpc 트래픽에 대해 기본 포트인 111번과 Solaris 2.6 이전 버전에서 사용하는 32771을 지정한다.



[그림 13] snort.conf의 기본 설정

* preprocessor bo

대표적인 Windows 기반의 해킹 툴인 Back Orifice를 탐지하기 위한 설정이다. 기본값을 지정하면 된다.



[그림 14] snort의 기본 설정

* preprocessor portscan: <감지할 네트워크대역> <포트수> <감지시간> <파일명>

snort의 강력한 기능중 하나인 포트 스캔이나 ip 스캔을 설정하는 부분으로 통상적으로 아래와 같이 설정하여 사용한다.

* preprocessor portscan: $HOME_NET 4 3 portscan.log

$HOME_NET 대역을 향하는 패킷에 대한 스캔을 감지하며 3초에 4개 이상의 포트에 접속을 요청할 때에는 스캔으로 인식하여 이 정보를 /var/log/snort/portscan.log에 저장한다는 의미이다. 만약 특정한 ip 에서의 스캔은 스캔으로 인지하지 않도록 설정하려면 아래의 preprocessor portscan-ignorehosts 설정을 이용하면 된다.

* preprocessor portscan-ignorehosts: 192.168.3.2 192.168.2.0/24

해당 ip를 지정하여 portscan을 탐지하지 않을 때 사용한다. 내부에 사용중인 nessus host와 같이 스캔 작업을 수행한다. 일반적으로 기본값을 사용한다.

* preprocessor sfportscan

sfportscan 기능은 sourcefire에서 개발되었으며, 포스 스캔이나 ip 스캔을 탐지하기 위해 설정하는 부분으로 아래와 같은 옵션을 이용하여 설정이 가능하다.

proto { tcp udp icmp ip_proto all } - 프로토콜 타입을 설정한다.

scan_type { portscan portsweep decoy_portscan distributed_portscan all } - 스캔 타입을 설정한다.

sense_level { low|medium|high } - 스캔의 위험도를 설정한다.

memcap { positive integer } - 스캔 탐지를 위한 할당된 최대 byte 값

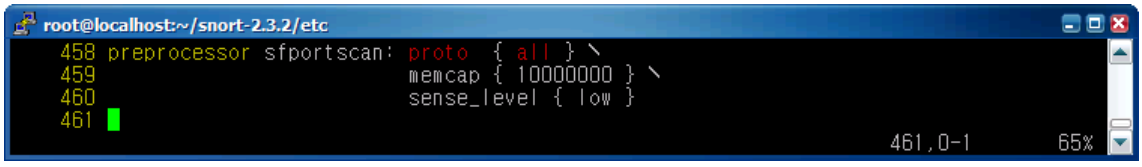
logfile { filename } - 로그 기록을 위한 파일 위치

watch_ip { Snort IP List } - 스캔이 자주 탐지되는 ip

ignore_scanners { Snort IP List } - 스캔을 탐지하지 않을 source ip (예 : nessus host)

ignore_scanned { Snort IP List } - 스캔을 탐지하지 않을 ip destination ip (예: syslog server)

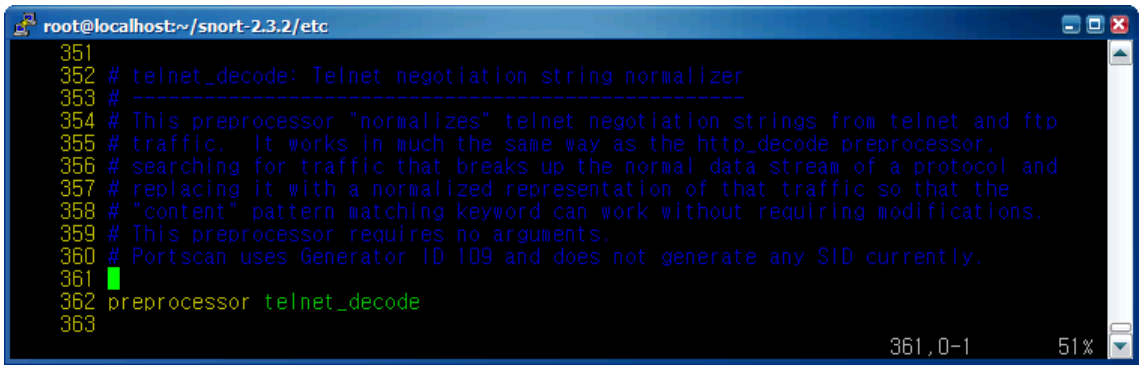
일반적으로 기본값을 사용해도 무방하다. 자세한 내용은 doc/README.sfportscan 을 참조한다.



[그림 15] snort.conf의 기본 설정

* preprocessor telnet_decode

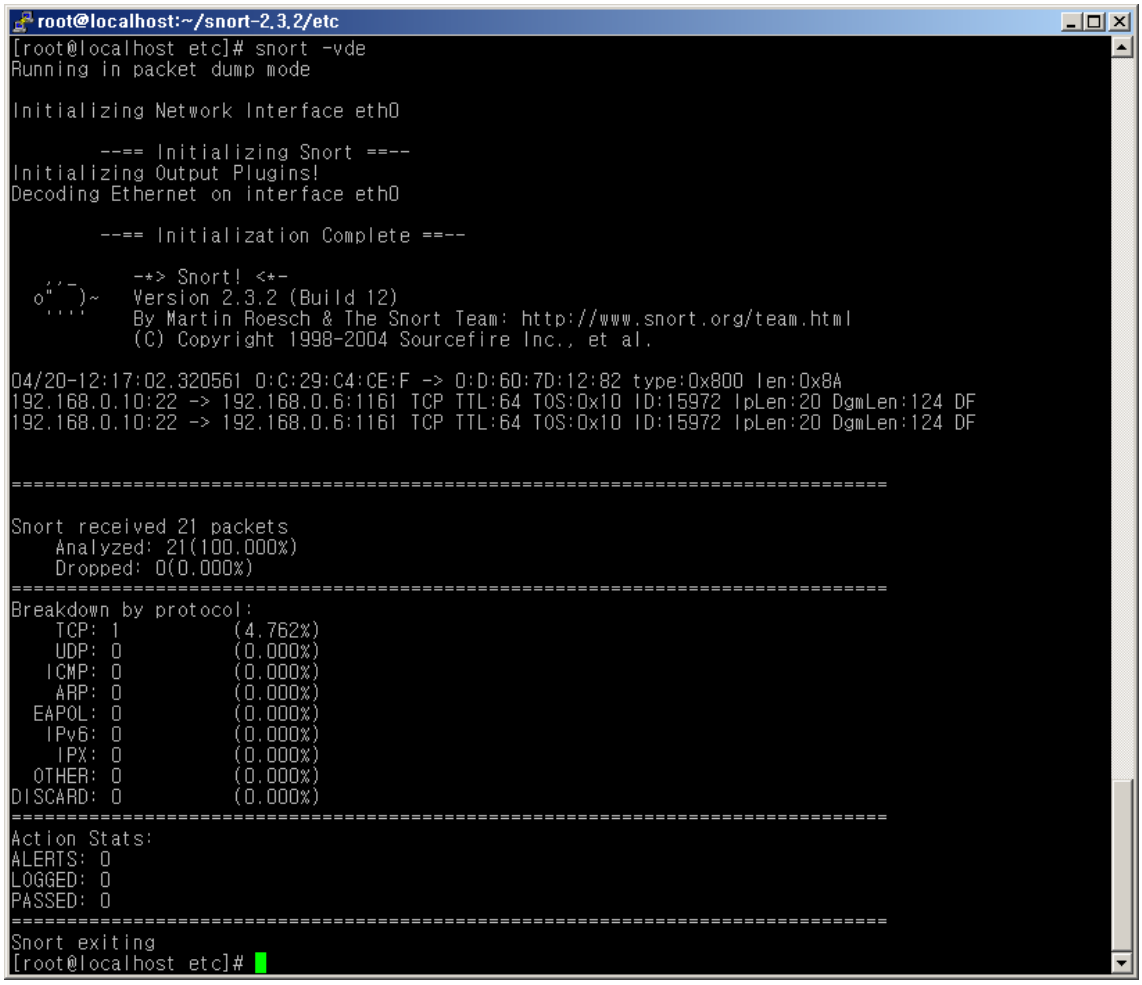
별도의 옵션 없이 기본값을 쓰면 된다.



[그림 16] snort의 기본 설정

STEP 4. snort 실행

snort -vde (snort 스니퍼 모드 실행)

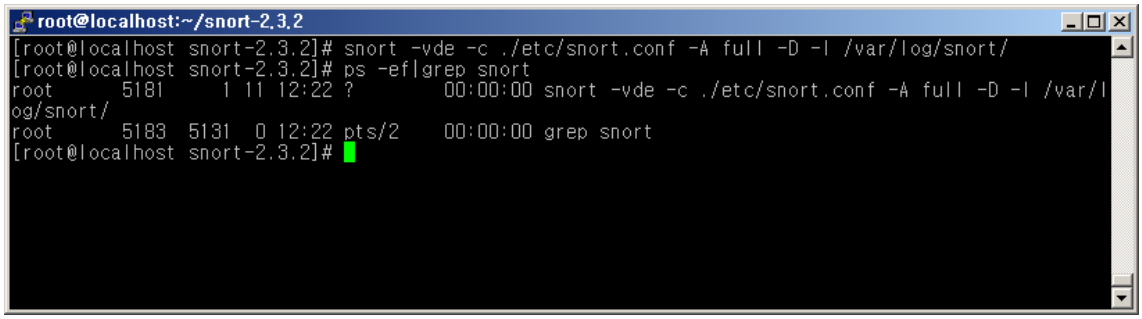


[그림 17] snort 기본 실행 화면

위의 실행 방법은 로그를 남기지 않고 실행하는 방법이다. 이는 실질적인 공격이 이루어지고 있다고 생각될 때, 실시간으로 확인하기 위해 가끔 사용하는 방법 중 하나이다.

snort -vde -c ./etc/snort.conf -A full -D -l /var/log/snort/

(snort 침입 탐지 모드 실행)



[그림 18] snort 실행

위의 실행 방법은 로그를 가장 많이 사용되는 침입 탐지 모드로 snort를 실행하는 방법으로 log 파일을 /var/log/snort/ 디렉토리에 저장하고, 데몬으로 snort를 실행하는 것이다.
log에 대한 자세한 설명은 <별첨 #1>, 각 모드에 대한 자세한 설명은 <별첨 #2>를 참조하도록 한다.

STEP 5. rule 자동 업데이트

룰 자동 업데이트는 snort에 기본적으로 제공되는 기능이 아니다. 이는 oinkmaster 라는 프로그램을 이용함으로써 가능하므로, 아래의 순서대로 따라 설치하면 쉽게 설치 및 실행이 가능하다. 룰에 대한 기본적인 내용은 <별첨 #3>을 참고하도록 한다.
snort 는 사전에 정의된 룰을 기반으로 비정상 트래픽인지 아닌지 여부를 판단하므로 새롭게 나오는 공격에 대해서는 새로운 룰을 적용하여야 한다. 따라서 Signature 기반의 IDS 인 snort를 운영할 때 rule 을 업데이트하고 각자의 상황에 맞게 수정하는 것은 가장 중요한 부분 중 하나라고 할 수 있는 것이다. 이를 통해 공격인데 공격으로 탐지를 못하거나 공격이 아닌데 공격으로 탐지하는 False positive 나 False negative와 같은 오탐을 줄일 수 있고, 하루가 다르게 새로운 양상을 띄는 최신의 공격 시도를 탐지할 수 있기 때문이다. 물론 초기에 snort를 설치하면 자체적인 룰을 제공하고 있기는 하지만 이것만으로는 부족한 것이 사실이다. 최신 탐지룰을 지원 받기 위해서는 최근 rule 배포 정책의 변경으로 인해 snort 홈페이지에 가입하여 rule 업데이트를 지속적으로 지원받을 수 있다.

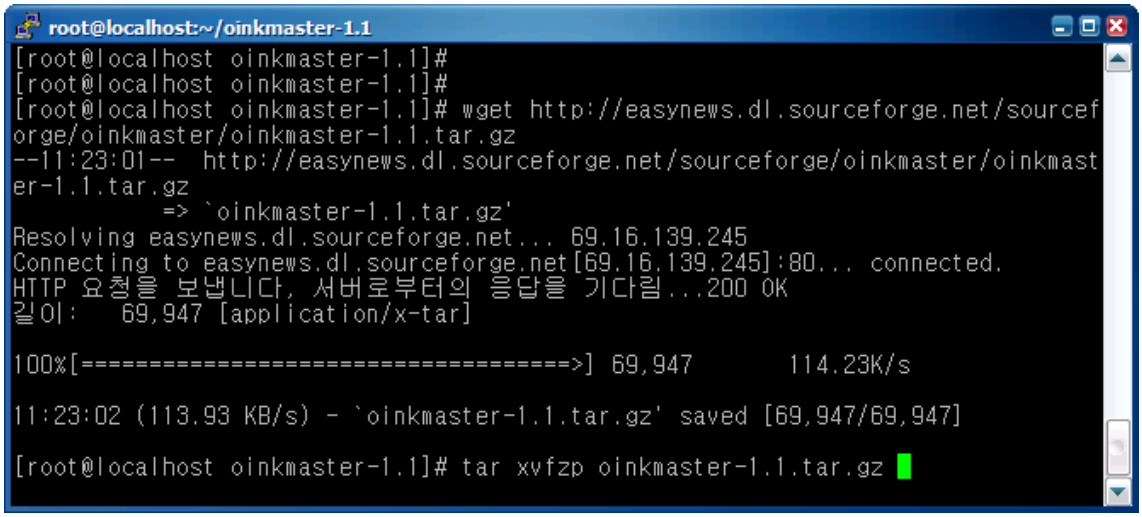
[그림 19] snort 홈페이지의 룰 다운로드 회원 가입

snort 탐지룰은 공식 사이트인 http://www.snort.org/rules/에서 다운로드 받을 수 있는데, 하루에도 수차례식 업데이트 되는 룰을 매번 다운로드 받는 것은 불가능할 것이다. 따라서 이러한 작업을 oinkmaster라는 perl 프로그램을 이용하면 자동화할 수 있어 업데이트로 인한 번거로움을 해결할 수 있다.

oinkmaster는 홈페이지인 http://oinkmaster.sourceforge.net/나 snort.org 에서도 다운로드할 수 있는데, 정상적으로 작동하려면 perl과 tar, gzip, wget 또는 Archive::Tar, IO::Zlib and LWP::UserAgent 모듈을 필요로 하므로 시스템에 설치되어 있는지 확인하기 바란다. (대부분 설치되어 있을 것이다.)

[그림 20] oinkmaster 홈페이지

먼저, 위 사이트에서 oinkmaster를 다운로드한다.



[그림 21] oinkmaster 다운로드 및 압축해제

실행파일은 oinkmaster.pl이고 설정파일은 oinkmaster.conf인데, oinkmaster.pl파일은 수정할 필요 없이 oinkmaster.conf 파일만 수정하면 된다.

```
url = http://www.snort.org/dl/rules/snortrules-snapshot-2_2.tar.gz
```

"url =" 은 룰 파일을 다운로드하여 업데이트 할 url을 지정하는 부분이다.
이전 버전에서는 위와 같이 설정 되어 있었지만 최근에 rule 배포 정책이 변경되어

무료 회원가입을 해야하며, 회원가입을 하면, 가입한 유저별로 업데이트할 url을 지정하여 알려준다. 회원 가입 후 등록한 메일로 아래와 같은 확인 메일을 수신한다.

제목 Snort.org Registration Information
보낸날짜 2005년 4월 4일 (月) 04:13 (-0400 (EDT))
보낸이 <registration@snort.org> » 수신거부 » 메일주소록에 추가
받는이

Thank you for registering at www.snort.org.
Your Snort account has been successfully created.
Please use the information below to login at www.snort.org.

User Name: user id
Password: user password

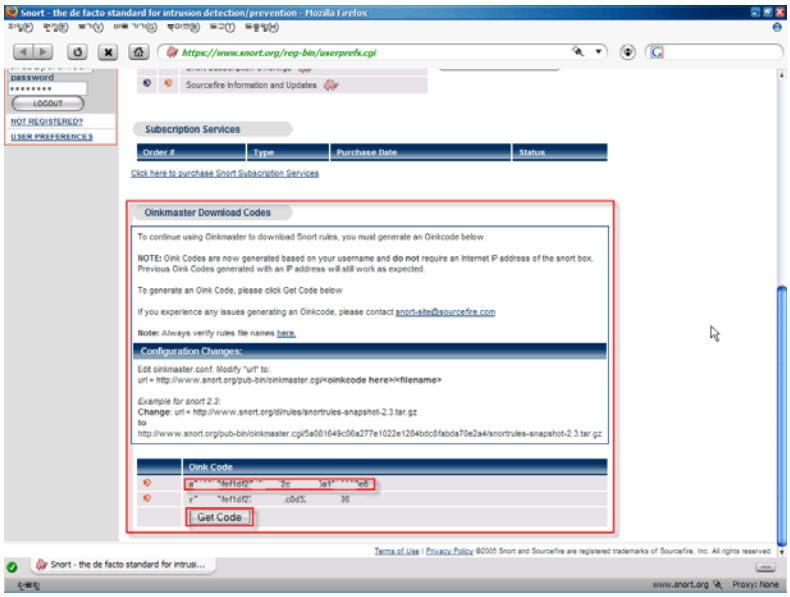
This password was randomly generated so, for security purposes,
we recommend that you change it when you first login. You can
do this at https://www.snort.org/reg-bin/userprefs.cgi

If you have any trouble logging in, please e-mail snort-site@sourcefire.com.

Thank you,
The Snort Team
www.snort.org

[그림 22] snort 홈페이지 가입 후 수신한 메일

수신한 메일에 id (메일주소)와 password (1회용 패스워드)가 포함되어 있으며, https://www.snort.org/reg-bin/userprefs.cgi로 접속하여 패스워드 변경 및 oinkmaster를 이용한 rule을 다운받을 수 있는 링크를 받을 수 있다.



[그림 23] onikmaster 다운로드 코드 획득

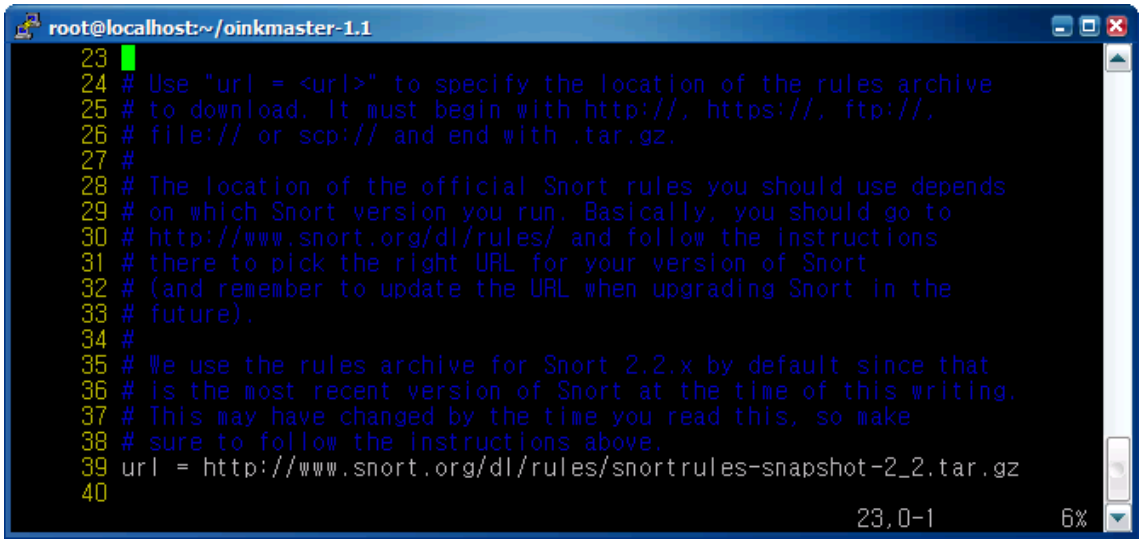
메일에 포함된 url로 이동하여, 보내온 id 및 password를 이용하여 로그인 하면 위와 같은 페이지에 접속이 되며, “Get Code" 버튼을 눌러 oink code를 가져올 수 있다. oink Code는 일반적으로 숫자와 영문 혼용으로 되어 있으며, 각 개별코드를 이용하여 아래와 같이 url을 생성할 수 있다.

```
http://www.snort.org/pub-bin/oinkmaster.cgi/[개별oinkCode]/snortrules-snapshot-2.3.tar.gz
```

예제) 생성한 oinkmaster url

```
http://www.snort.org/pub-bin/oinkmaster.cgi/5a0814284bdc8fabda70e2a4/snortrules-snapshot-2.3.tar.gz
```

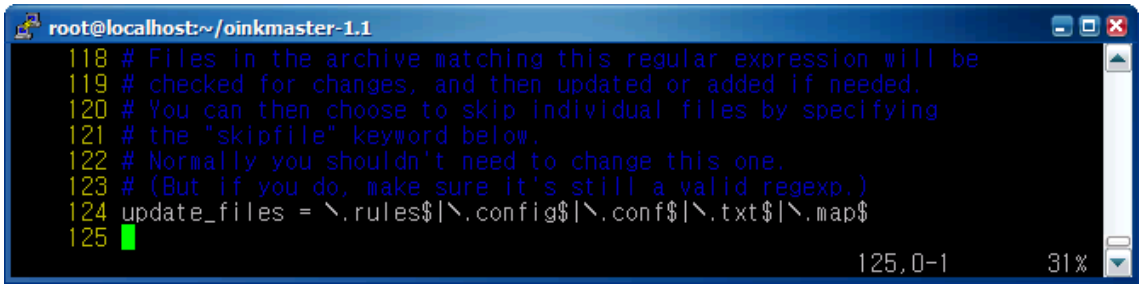
위에서 획득한 oinkmaster url을 onikmaster.conf 파일에 입력하여 작성한다.



[그림 24] oinkmaster.conf의 기본 설정

```
update_files = \.rules$|\.config$|\.conf$|\.txt$|\.map$
```

이 부분은 어떤 확장자를 가진 파일을 업데이트 할 것인지 지정하는 것인데, 변경할 필요 없이 기본값을 그대로 사용하면 된다.

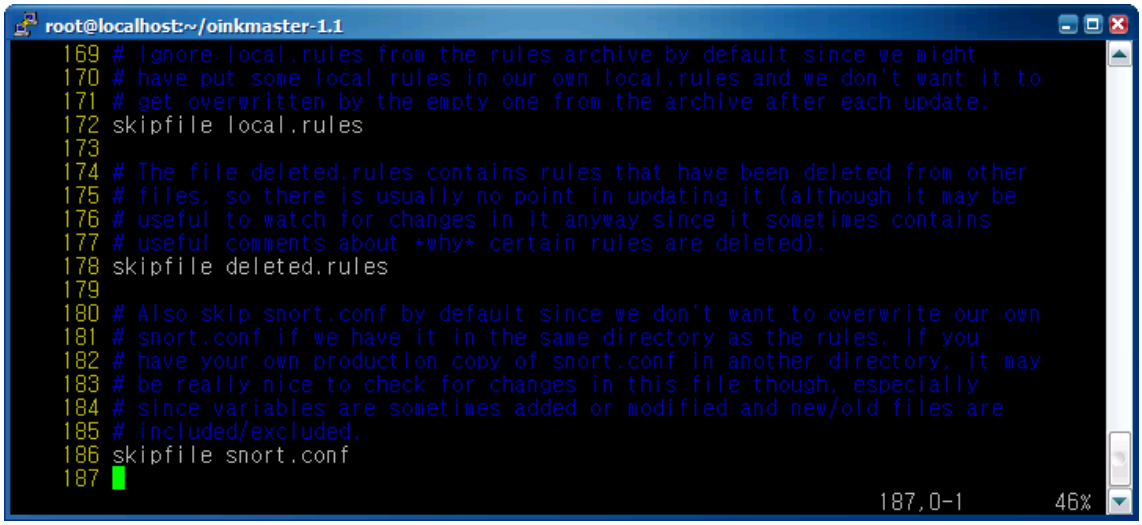


[그림 25] oinkmaster.conf의 기본 설정

```
skipfile local.rules
skipfile deleted.rules
skipfile snort.conf
```

skipfile은 업데이트시 덮어쓰지 않을 파일을 지정하는 것으로 위 두 줄은 반드시 설정되어 있어야 한다. local.rules는 각자의 환경에 따라 룰을 커스터마이징하기 위

해 설정을 추가하는 룰 파일이고, snort.conf는 ip 주소등이 설정되어 있는 주 설정 파일이다.

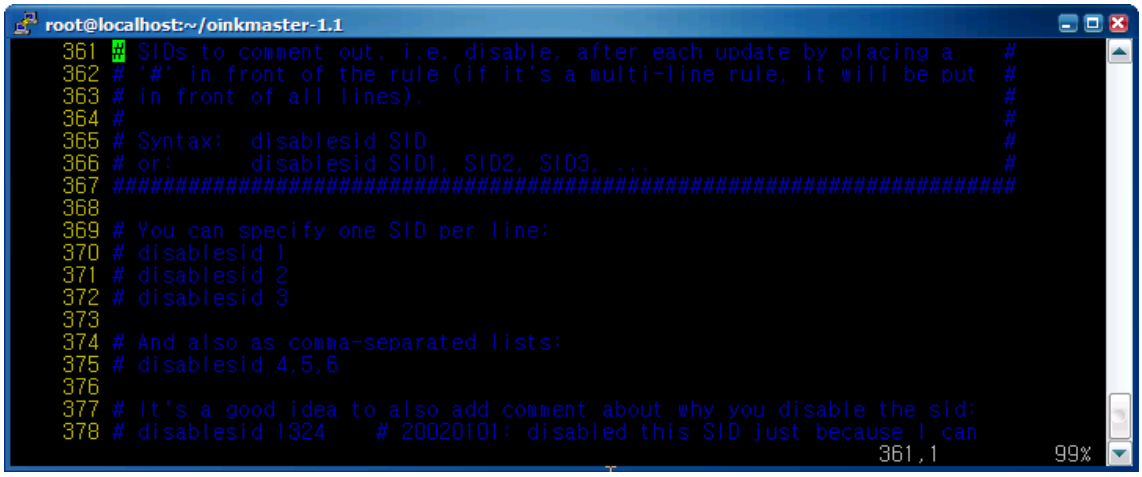


[그림 26] oinkmaster.conf의 기본 설정

```
disablesid 1
disablesid 2
disablesid 3
```

이 부분은 oinkmaster를 통해 룰 업데이트시 업데이트를 하지 않을 룰 번호(sid)을 지정하는 부분이다. 실제로 snort를 운영하다 보면 자신의 환경에 맞지 않거나 불필요한 룰이 있는 경우가 있다. 이를테면 NMS나 SMS에서 모니터링을 위해 각 서버나 네트워크에 접속 시도를 하게 되는데 이를 스캔 공격으로 인식하는 것이 그 예가 될 수 있다. 이러한 경우에는 경고를 출력하는 해당 룰을 주석처리하거나 수정하면 되는데, 만약 oinkmaster를 이용하여 업데이트를 하게 되면 룰 설정이 초기화되어 설정했던 주석이 초기화되는 경우가 있으므로 미리 지정된 특정한 룰에 대해서는 업데이트 하지 않도록 위와 같이 disabled 뒤에 해당 룰의 sid 번호를 설정해 두면 된다. 만약 disable 할 룰이 많을 경우에는 아래와 같이 , 를 기준으로 지정할 수도 있다.

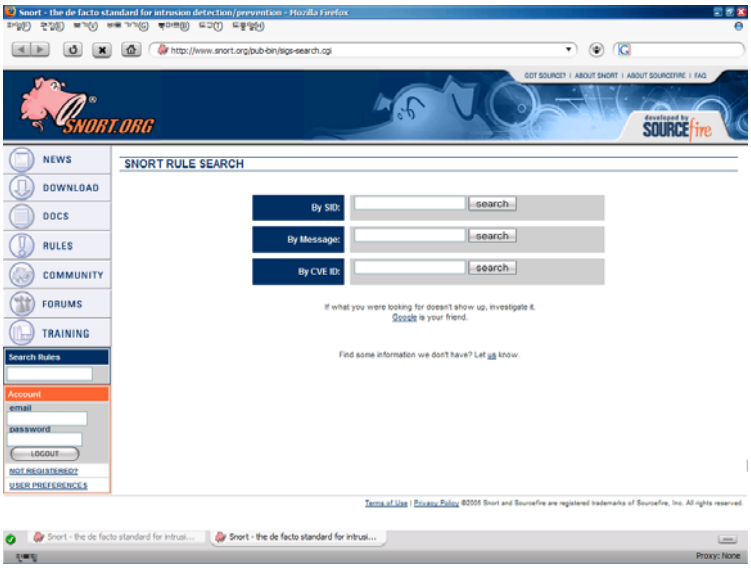
```
disablesid 1, 2, 3, 451, 21, 51
disablesid 1324
```



[그림 27] oinkmaster.conf의 기본 설정

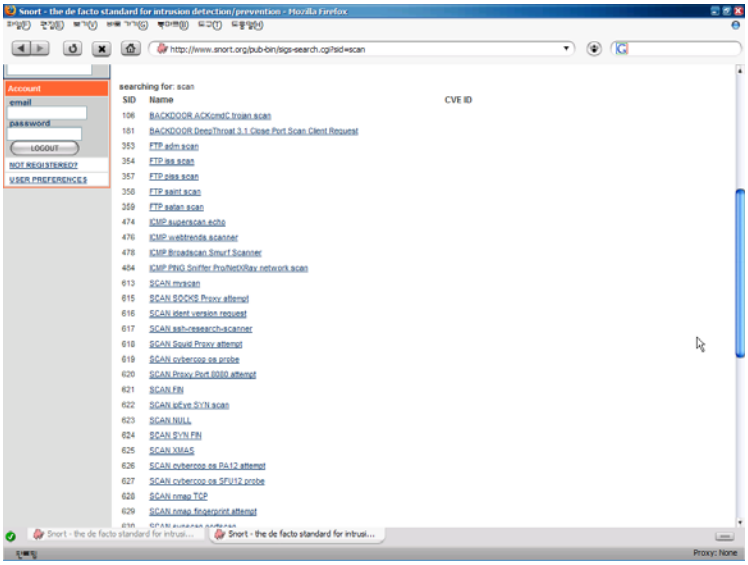
<TIP> 특정한 룰의 sid 확인

만약 msg 에 'SCAN Proxy' 라는 문자열이 포함된 룰을 찾으려면 http://www.snort.org/pub-bin/sigs-search.cgi 에 접속하여 By Message 부분에 검색하고자 하는 메시지를 입력 후 조회해 보면 된다. 또는 반대로 By SID 부분에 룰의 sid를 입력하면 룰의 상세한 내용을 조회할 수 있다.



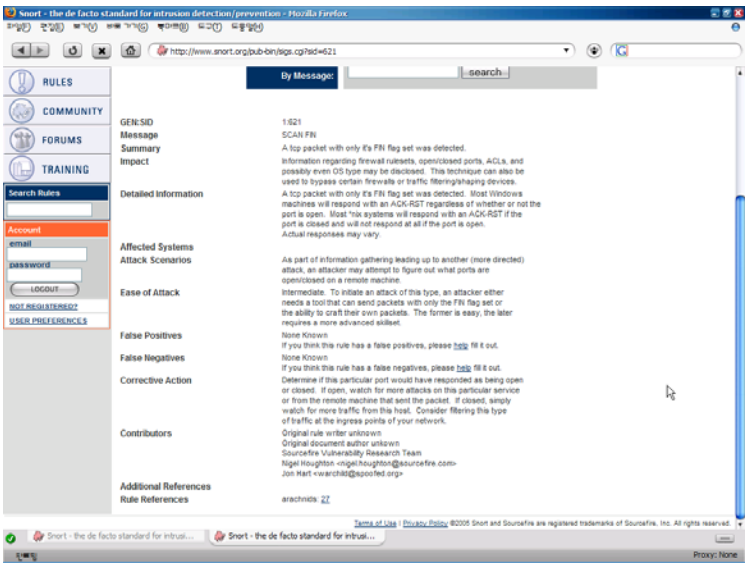
[그림 28] Message 로 SID 찾는 URL

아래는 Message에 SCAN으로 입력하였을 때 보이는 결과 화면이다. 이는 msg에 SCAN이 포함된 룰의 sid와 룰이름을 보여주고 있다.



[그림 29] SCAN 으로 질의한 결과

이후 해당 룰을 클릭하면 아래와 같이 룰에 대한 상세한 정보가 설명되어 있어 해당 룰이 탐지하는 공격 또는 트래픽에 대해 좋은 자료를 제공하고 있다.



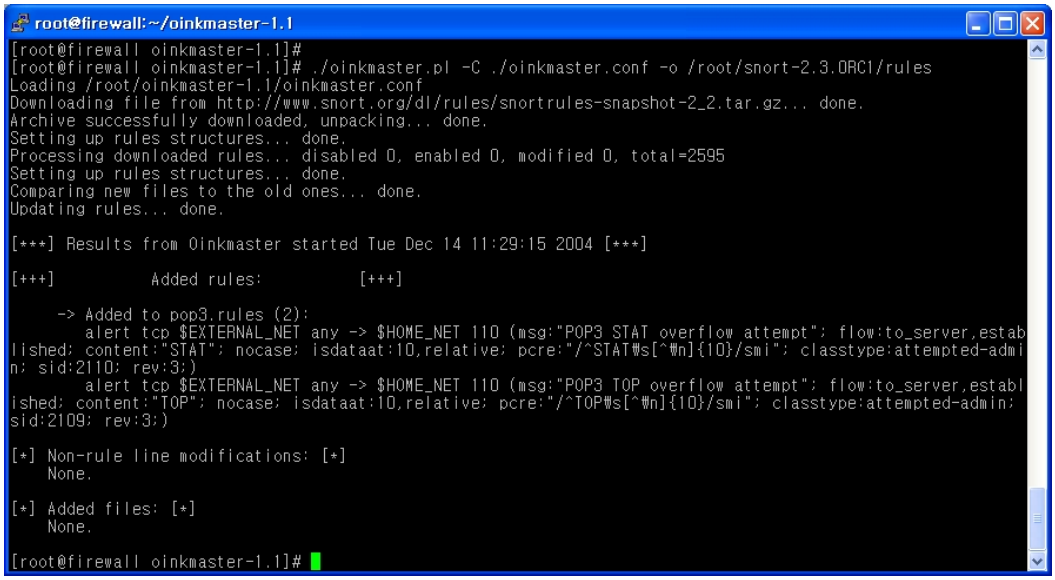
[그림 30] 특정 sid 에 대한 자세한 설명

이제 oinkmaster.pl을 실행하여 룰을 업데이트 해 보자.

oinkmaster.pl 에서 제공하는 옵션은 몇 가지가 있는데, 간단히 아래와 같이 실행하면 된다.

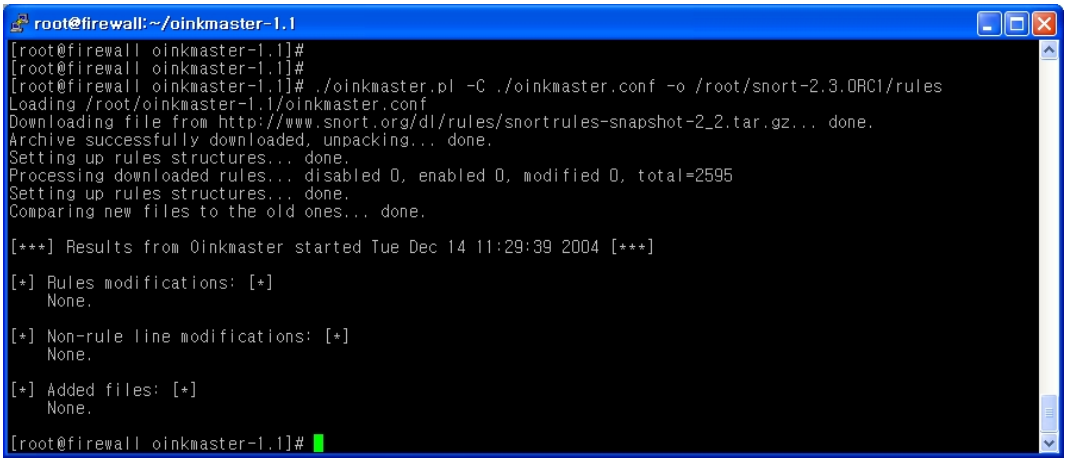
#./oinkmaster.pl -C “oinkmaster.conf의 파일위치” -o “rules 디렉토리 경로”

아래 그림의 경우 실행을 해 보면 먼저 snort 홈페이지에서 최신의 룰 파일을 다운로드 한 다음 압축 해제 후 -o “rules 디렉토리 경로”에서 지정한 경로에 있는 파일과 비교를 하게 된다. 그리고 변경하거나 추가된 파일 없이 2개의 룰을 추가하였다는 것을 알 수 있다.



[그림 31] oinkmaster 실행 화면

아래 그림의 경우 같은 방법으로 실행되었지만 최신의 룰 업데이트가 되어 추가, 변경된 룰이 없다는 것을 보여주고 있다.



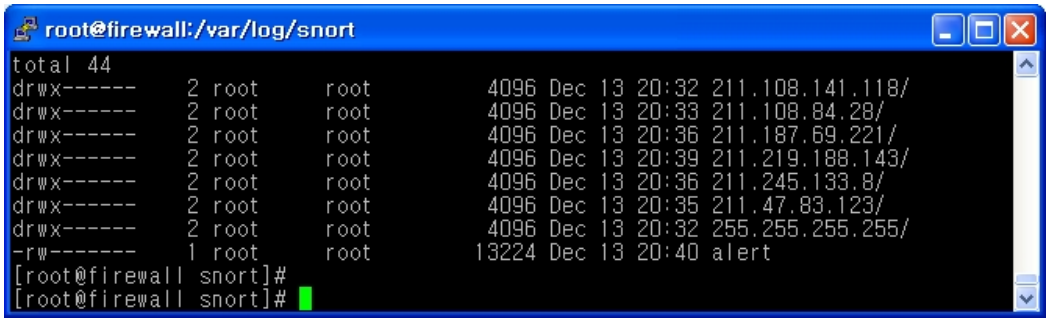
[그림 32] oinkmaster 실행 화면

이제 정상적으로 룰 업데이트를 완료하였는데, 위 명령어를 /etc/cron.daily/ 와 같이 cron에 설정하여 매일 정기적으로 업데이트 하도록 설정하면 될 것이다.

이상으로 snort에 대한 기본적인 설치 및 실행, 탐지 룰 업데이트에 대한 내용을 살펴보았다. snort는 침입탐지를 위해 다양한 응용 프로그램 및 plugin 등이 각 오픈소스 개발자들 사이에서 작성되고 있으며, snort 관리에 유용한 프로그램도 다수 존재한다. snort의 다양한 plugin 및 관리 프로그램에 대한 예제는 <별첨 #4>에서 확인하면 된다.

<별첨 #1> snort 로그의 다양한 저장법

만약 모니터링하는 패킷이 룰에 매칭되었을 경우 어떤 방법으로 로그에 남길 것인지 지정하는 부분으로 시스템의 /var/log/ 디렉토리 이하에 로그 파일을 생성하는 방법과 database 에 저장하는 방법, 그리고 binary tcpdump 형식으로 남기는 방법 등이 있다. 가장 많이 사용되는 방법으로서 아무런 로깅 옵션을 설정하지 않으면 /var/log/snort 디렉토리에 탐지 관련 파일들이 생성되는데, 아래는 실제 생성된 예이다.



[그림 33] snort 실행으로 생성된 로그

위에서 모든 경고는 alert 라는 파일에 순서대로 요약되어 저장되고, 룰에 탐지된 패킷의 소스 ip 를 디렉토리 이름으로 한 하위 디렉토리가 생성되고 이 디렉토리 안에 상세한 패킷 정보가 저장된 파일이 생성된다. 아래는 /var/log/snort 디렉토리에 생성된 alert 파일의 예이다.

```
[**] [1:469:3] ICMP PING NMAP [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/13-20:33:05.873163 211.108.84.28 -> 221.1.2.3
ICMP TTL:119 TOS:0x0 ID:42135 IpLen:20 DgmLen:28
Type:8 Code:0 ID:512 Seq:38139 ECHO
[Xref => http://www.whitehats.com/info/IDS162]
```

```
output database: log, mysql, user=root password=test dbname=db host=localhost
output database: alert, postgresql, user=snort dbname=snort
output database: log, unixodbc, user=snort dbname=snort
output database: log, mssql, dbname=snort user=snort password=test
```

로그 정보를 /var/log/snort 에 남기지 않고 database 에 기록하는 방법이 있다. snort에서 지원하는 db 는 Postgresql, MySQL, unixODBC, MS-SQL Server 와 Oracle 등이다., 가장 많이 사용되는 mysql을 예로 들어 설정해 보도록 하겠다. 참고로 만약 snort 의 로그를 mysql 과 같은 db 에 저장하려면 설치 과정에서 configure 시 아래와 같이 별도의 옵션을 명시해 주어 설치하여야 한다.

```
./configure --with-mysql=/usr/local/mysql; make; make install
```

db를 이용한 로깅에 대한 자세한 설명은 doc/README.database 파일을 참고하기 바란다.
etc/classification.config 는 각 침입탐지 룰의 class 타입과 중요도를 지정하며
etc/reference.config 는 각 취약성에 대한 참조 URL을 정의하는 것으로 기본설정대로 include 하도록 한다.

<별첨 #2> snort 실행 모드

snort 는 단순한 패킷을 캡처하는 스니퍼 모드로 사용될 수도 있고 룰을 매칭함으로써 침입 탐지 모드로도 사용될 수 있다고 하였는데, 각각에 대해 알아보도록 하자.

① 스니퍼 모드로 실행하기
기본적인 스니퍼 또는 dump 모드를 사용하기 위해서는 아래와 같이 간단히 -v 옵션을 주어 실행하면 아래와 같이 패킷의 정보가 출력된다.

위의 정보를 보면 192.168.68.252에서 192.168.68.255 으로 UDP 패킷이 전송된 것을 알 수 있다. 그리고 패킷의 TTL 이나 TOS, ID, IpLen 등이 자세하게 보여진다.이후 Ctrl + C 로 작동을 멈추면 보여진 패킷에 대하여 tcp, udp, icmp 등에 대한 간단한 통계를 보여준다. 이 모드는 비정상적인 패킷에 대한 침입을 탐지하는 것이 아니며 단지 snort에서 보이는 모든 패킷에 대한 정보를 출력한 것뿐이다. 따라서 이 모드로 실행시 보이는 정보는 정상 패킷일 수도 있고 비정상일 수도 있는 것이다.
방금 실행한 -v 옵션과 함께 좀 더 상세한 정보를 보고자 한다면 다음의 두 옵션을 추가하여 사용할 수 있다.
-d : 응용프로그램 수준(Application Layer)의 데이터를 HEX 나 ASCII 모드로 보여준다.
-e : MAC 주소 등의 데이터도 보여준다.

아래는 “snort -vde” 를 실행하였을 때 보이는 정보로서 첫줄에는 MAC 정보가, 두 번째 줄에는 ip 정보가 보인다. 그리고 우측에는 ASCII 가 디코딩된 정보가 출력되며 이를 통해

```
02/19-16:17:56.790278 0:2:FC:8:C4:A0 -> 0:50:8B:9A:1B:1B type:0x800 len:0x3C
10.2.3.4:110 -> 10.2.4.39:1763 TCP TTL:121 TOS:0x0 ID:22164 IpLen:20 DgmLen:40
***AP*** Seq: 0x2B9D2415  Ack: 0x6405A45C  Win: 0x16D0  TcpLen: 20

2B 4F 4B 20 50 4F 50 33 20 73 74 61 66 66 73 2E  +OK POP3 mail.
74 74 2E 63 6F 2E 6B 72 20 76 32 30 30 31 2E 37  domain.co.kr v2001.7
38 72 68 20 73 65 72 76 65 72 20 72 65 61 64 79  8rh server ready
0D 0A
```

mail 서버에 pop3 접속한 것을 알 수 있다.

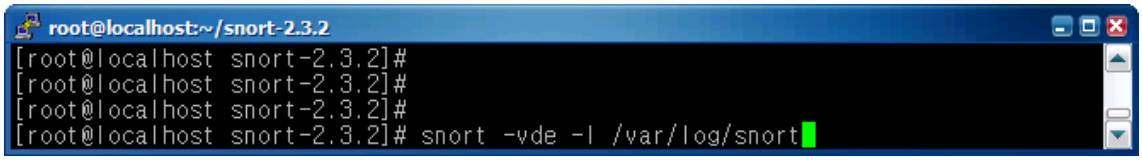
또한 snort 는 BPF를 제공하여 tcpdump 처럼 패킷캡처를 원하는 구체적인 설정을 할 수 있다. 이를테면 pop3 와 관련된 트래픽만 캡처하고자 할 경우에는 “snort -vde port 110"을 실행하면 될 것이고, http 와 관련된 트래픽만 캡처하고자 한다면 "snort -vde port 80"을 실행하면 될 것이다. 만약 ip 가 1.2.3.4 와 관련된 트래픽만 캡처하고자 한다면 ”snort -vde host 1.2.3.4"를 실행하면 되고 만약 udp 만 캡처하고자 한다면 “snort -vde udp"를 실행하면된다

<참고> BPF
BPF(Berkeley Packet Filter)는 스니퍼나 네트워크 모니터링 같은 패킷 캡쳐 프로그램에 표준적으로 적용되는 필터를 칭하며, 문자적인 표현이 많아서 이해하기가 쉬운 장점이 있다.

```
[root@firewall snort-2.3.0RC1]# ./snort -v
Running in packet dump mode
Log directory = /var/log/snort

Initializing Network Interface eth0

           --== Initializing Snort ==--
Initializing Output Plugins!
Decoding Ethernet on interface eth0

           --== Initialization Complete ==--

    ,,_       -*> Snort! <*-
   o"  )~     Version 2.3.0RC1 (Build 8)
    ''''      By Martin Roesch & The Snort Team: http://www.snort.org/team.html
              (C) Copyright 1998-2004 Sourcefire Inc, et al.



12/13-21:37:24.479387 192.168.68.252:137 -> 192.168.68.255:137
UDP TTL:128 TOS:0x0 ID:49566 IpLen:20 DgmLen:78
Len: 50


===============================================================================
Snort received 35 packets
      Analyzed: 35(100.000%)
      Dropped: 0(0.000%)
===============================================================================
Breakdown by protocol:
      TCP: 29              (82.857%)
      UDP: 2               (5.714%)
     ICMP: 2               (5.714%)
      ARP: 0               (0.000%)
   EAPOL: 0                (0.000%)
    IPv6: 0                (0.000%)
      IPX: 0               (0.000%)
   OTHER: 0                (0.000%)
DISCARD: 0                 (0.000%)
===============================================================================
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
===============================================================================
```

② log 파일로 저장하는 방법

위와 같이 snort를 실행하면 너무 많은 패킷이 화면에 보여지기 때문에 현실적으로 어떤 내용으로 캡처되는지 알 수 없다. 따라서 snort 의 실행 결과를 파일로 저장하도록 할 수 있는데, 이를 위해서는 아래와 같이 log의 의미인 -l 을 지정하고 로그를 남길 디렉토리를 지정하기만 하면 된다.



[그림 34] log 파일로 저장하는 설정

잠시 후 /var/log/snort 디렉토리를 살펴보면 소스 ip 인 211.219.188.143 을 이름으로 한 디렉토리가 생성되고, 그 디렉토리에 다음과 같은 파일이 생성된 것을 알 수 있다.

[그림 35] 생성된 로그 파일

파일 이름을 보면 해당 패킷은 ICMP 이며, ICMP_ECHO 인 것을 알 수 있다.

③ 침입탐지(IDS) 모드로 실행하는 방법

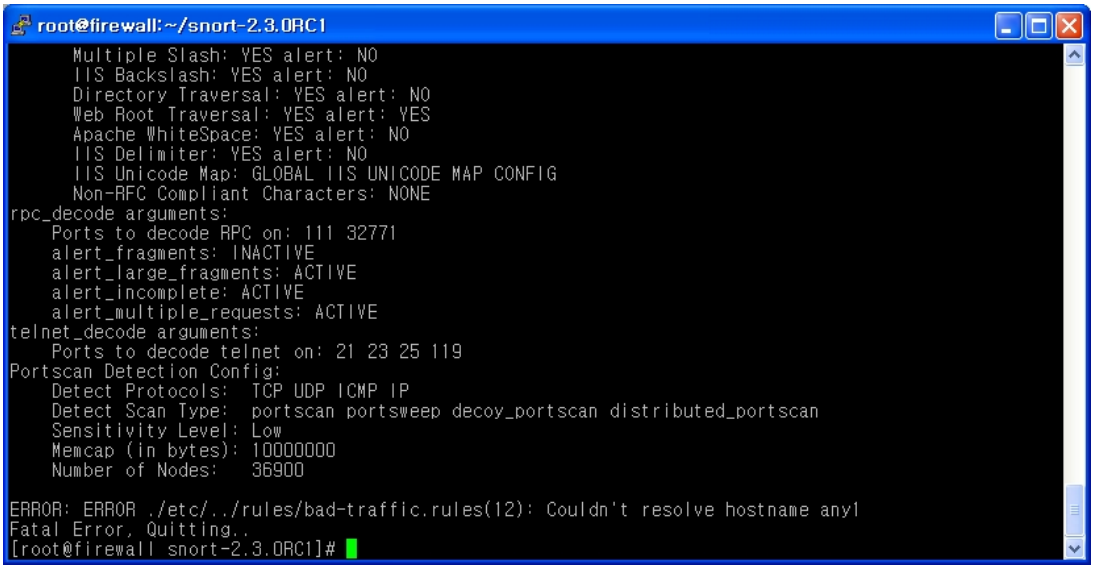
다음으로 snort 의 핵심 기능은 침입탐지 모드에 대해 알아보자. 침입탐지 모드는 경계 모드와 로깅 모드가 있는데, 경계 모드에서는 snort 구동시 -A 옵션을 주어 지정할 수 있으며 Full 과 Fast가 올 수 있다. 여기에서 Full 은 기본 옵션으로 alert 뿐만 아니라 모든 패킷 헤더 정보를 출력하게 된다. 그리고 Fast 는 시간이나, 경고메시지, 소스/목적지 ip 및 포트 등 단순한 형식으로 저장하게 된다.

이외 snort에서 제공하는 옵션이 많이 있는데, 통상적으로 다음과 같이 사용하면 무난하다.

```
# snort -d -c /usr/local/snort/etc/snort.conf -A full -D
```

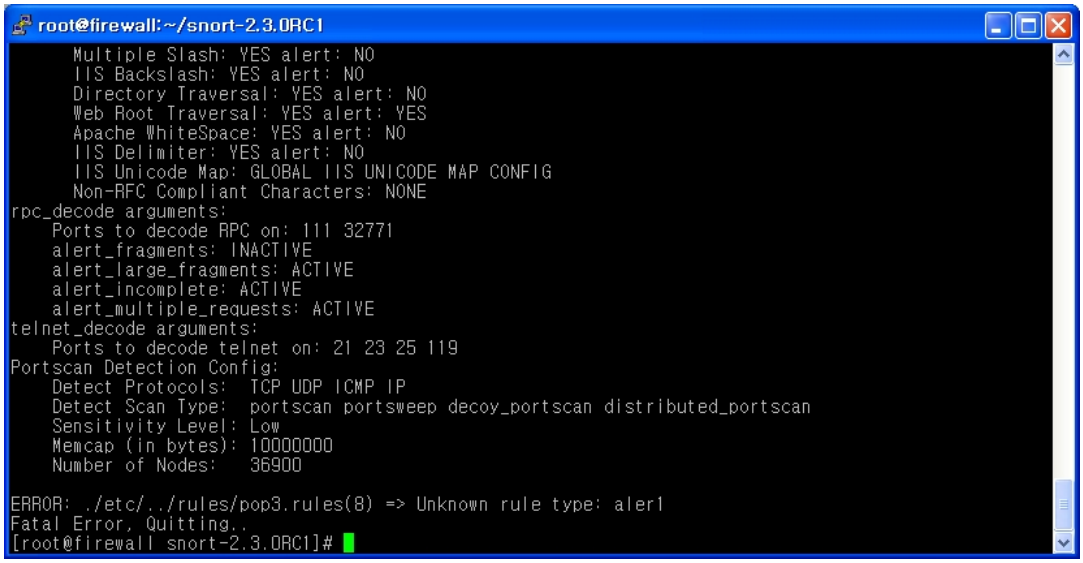
여기에서 -c 는 config 의 의미로서 snort 설정파일인 snort.conf 파일의 위치를 지정하고, -D 는 데몬 모드로 시작한다는 의미이다.

만약 snort를 구동시 설정 파일에 문법적인 오류가 있을 경우에는 제대로 작동하지 않으므로 구동 전에 snort -c /usr/local/snort/etc/snort.conf를 실행하여 문법적인 오류가 없는지 여부를 먼저 확인해 보아야 한다.



[그림 36] snort.conf 에러시 화면

위의 경우 ip 설정 부분에서 any 가 아니라 any1 로 오타가 있어 에러가 나면서 실행이 종료된 것을 알 수 있다.



[그림 37] snort.conf 에러시 화면

위의 경우 pop3.rules 파일에 alert 가 아니라 aler1 으로 오타가 있어 실행시 에러가 나는 것을 알 수 있다.

<별첨 #3> snort 룰 이해하기

룰 파일은 rules 디렉토리 밑에 공격의 여러 가지 파일로 분류되어 있는데, 해당 룰을 이용하려면 include 형식으로 해당 룰 파일을 참조하도록 하면 되고 사용하지 않으려면 해당 include 앞에 주석처리를 하면 된다.
rule 파일은 다음과 같이 크게 header 와 body 로 구분되며, 아래의 예는 snmp(161/udp) community 문자열로 public 이 설정된 패킷을 탐지하는 예이다.

```
# 헤더
alert udp $EXTERNAL_NET any -> $HOME_NET 161

# body
(msg:"SNMP public access udp"; content:"public"; reference:cve,CAN-1999-0517; reference:cve,CAN-2002-0012; reference:cve,CAN-2002-0013; sid:1411; rev:3; classtype:attempted-recon;)
```

위의 예제 룰을 참고로 룰의 헤더에 들어갈 수 있는 경우를 알아보자.
① alert : 룰의 action 에는 주로 alert 가 사용되는데 이외에도 pass 나 log 등이 사용될 수 있다. 여기에서 pass 는 해당 룰을 매칭되는 패킷에 대해 그대로 통과시키고자 할 때 사용되는데, 특히 특정한 ip 에 대해서 룰이 매칭될 때 로그 파일을 남기자 않고 통과시키고자 할 때 사용된다. log 는 지정한 룰에 매칭되는 정보를 /var/log/snort 디렉토리에 소스 ip를 디렉토리 이름으로 한 상세한 로그 정보를 남기는 것은 동일하지만 alert 파일에는 남기지 않는다는 차이점이 있다.

② udp : 룰에서 매칭하려는 프로토콜(protocol)을 명시하는 부분이다. 이외 tcp 나 icmp 또는 ip 가 올 수 있다.

$EXTERNAL_NET : 이 부분은 앞에서도 살펴보았듯이 소스 ip 또는 ip 대역을 지정하는데, 각 룰마다 구체적인 룰 설정을 할 수 있다.

③ any : 소스포트를 지정하는데 소스포트의 경우 통상적으로 특별히 지정된 포트가 없으므로 any를 사용한다.

④ ->: 화살표는 패킷의 방향을 의미한다. 위의 경우 $EXTERNAL_NET에서 $HOME_NET의 161/udp 로 향하는 패킷을 의미한다.

⑤ $HOME_NET : 목적지 ip를 지정하는 부분이다. 여기에서는 snort.conf 에서 지정한 $HOME_NET을 변수로 지정하였는데, 각 룰마다 구체적인 룰 설정을 할 수 있다.

⑥ 161 : 패킷의 목적지 포트를 지정한다. 만약 any 로 지정하면 모든 포트를 의미하게 된다.

다음으로, rule 의 body 에 들어갈 수 있는 경우를 살펴보자.

① msg : 이 부분은 해당 룰에 매칭되었을 때 alert 파일에 표시될 부분이다.
이를테면 룰 파일에 아래와 같이 지정되어 있을 경우

```
msg:"SNMP request udp";
```

alert 파일을 보면 아래와 같이 보이게 된다.

```
[**] [1:1417:9] SNMP request udp [**]
```

② content: 이 부분은 패킷에 어떤 문자열이 보을 경우 룰에 매칭하는지에 대한 설정으로 위의 경우 “public” 이라는 문자열을 포함하고 있을 경우 매칭한다는 의미가 된다. content 는 binary 형식일 수도 있고 text일 수도 있으며 두 개가 복합된 형식일 수 도 있다.

③ reference : 지정한 룰에 매칭 되었을 경우 비정상적인 트래픽에 대한 상세한 정보 제공을 위해 reference에서 지정한 URL 이나 정보가 함께 출력된다. 위의 경우 생성된 alert 파일을 보면 아래와 같은 URL 이 보이게 된다.
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0013

④ sid : sid 는 snort rule에서 사용하는 각 룰의 id 번호로 각 룰마다 고유한 번호가 사용된다. 룰에 대한 정보는 아래 URL에서 제공하고 있다.
http://www.snort.org/snort-db

⑤ rev : 이는 룰의 버전이라고 생각하면 된다. 즉, 같은 제목의 룰이라도 버전에 따라 룰이 다를 수 있는데, 당연히 rev 가 높은 버전의 룰을 사용하는 것이 좋을 것이다.

⑥ classtype : 공격 유형에 대한 분류 및 정보로서 이 정보를 통해 공격에 대해 쉽게 파악할 수 있으며 더불어 각 침입에 대한 중요도를 수치로 나타내는데, 1은 매우 높고, 2는 중간, 3은 매우 낮은 것이다.

지금까지 snort.conf 의 설정에 대해 알아보았는데, 일단 실행을 해 보고 생성되는 로그 및 탐지 결과등을 보면서 자신의 환경에 맞게 적절히 수정하면서 사용하면 될 것이다. 다음 절에서는 설정 파일을 근거로 직접 snort를 실행해 보도록 하자.

<별첨 #4> snort에 관련된 유용한 프로그램 소개

가. 웹기반의 IDS 모니터링 시스템 구축하기

snort 를 실행하는 방법도 알았고 룰 업데이트도 매일 자동으로 실행하도록 설정한 후, 생성되는 침입탐지 로그를 보면서 발견되는 오탐은 적당히 커스터마이징을 하면 더욱 효율적이다. 그런데, 시스템에 생성되는 침입탐지 로그를 매번 원격으로 로그인해서 일일이 로그파일을 보는 것이 여간 번거로운 일이 아닐 수 없다. 이를 위해서 별도의 웹 기반 프로그램을 이용하면 저장된 결과도 쉽게 모니터링이 가능하고 통계등의 정보도 볼 수 있을 것이다. 이러한 프로그램중에는 php 기반의 프로그램인 ACID(Analysis Console for Intrusion Databases) 가 가장 대표적으로 사용되는데, 이 프로그램을 이용하여 다양한 질의 및 검색 기능을 이용할 수 있고 경고창에 뜬 로그 정보를 가지고 디코딩된 패킷등도 모니터링 가능하다. 또한 각종 이벤트에 대하여 시간대별, 프로토콜별, ip 별등 다양한 기준으로 통계도 확인할 수 있다. ACID 는 http://acidlab.sourceforge.net/에서 다운로드하여 사용하면 된다. ACID 의 주 언어는 php 이므로 php 만 작동하는 OS라면 리눅스 뿐만 아니라 FreeBSD 나 Winodws 에서도 실행된다. 여기에서는 기본적으로, 리눅스 환경에서 사용한다고 가정하고 설치해 보도록 하자. 먼저 ACID를 이용하여 snort IDS를 구축하기 위해서는 아래와 같은 프로그램이 필요하다.

* Libpcap : snort 가 네트워크 인터페이스를 통해 패킷을 캡처하고자 할 때 패킷캡처 라이브러리인 Libpcap 를 필요로 한다.

* Mysql : mysql 은 적은 용량으로도 탁월한 성능을 제공하는 DBMS 로서 snort를 통해 수집된 로그는 매니저 서버의 mysql 에 DB 형태로 저장된다.

* Apache : 매니저 서버에서 분석 결과를 모니터링하고 관리하기 위해 필요하다.

* PHP: DB에 저장된 snort 로그는 매니저 서버에서 PHP 언어를 이용하여 분석한다. PHP 는 4.x 이상 버전을 사용하도록 한다.

* ACID : ACID는 Analysis Console for Intrusion Detection 의 약자로 PHP 언어로 짜여져 있으며 snort 의 로그를 체계적이고 유연하게 보여 질 수 있도록 구성된 프로그램이다.

* 기타 : ADODB 는 PHP4에서 필요로 하는 라이브러리이며 Zlib는 압축 라이브러리, JPGraph 는 PHP에서 그래프를 그리는데 필요하다. 이 패키지는 ACID에서 통계등의 결과를 그래프로 확인할 때 필요하다.

위에서 언급한 각각의 프로그램은 아래의 url에서 최신 버전을 다운받아 설치하도록 한다.
* MySQL
  http://www.mysql.com/

* Snort
  http://www.snort.org/
* apache
  http://www.apache.org/
* PHP
  http://www.php.net/
* ADODB
  http://phplens.com/lens/dl/adodb330.tgz
* Acid
  http://acidlab.sourceforge.net/
* Zlib
  http://www.gzip.org/zlib/
* JPGraph
  http://www.aditus.nu/jpgraph/
* LibPcap
  http://www.tcpdump.org/

mysql등 각각의 설치 방법은 각 소스의 README 나 INSTALL 파일을 보고 설치하기 바라며 이외 몇 가지만 살펴보도록 하자.

* zlib 1.1.4 설치:

```
# tar -xvzf zlib-1.1.4.tar.gz
# cd zlib-1.1.4
# ./configure; make install
```

* JPGraph 1.11설치:

```
# cp jpgraph-1.11.tar.gz /usr/local/apache/htdocs
# cd /usr/local/apache/htdocs
# tar xvzf jpgraph-1.11.tar.gz
```

* ADODB 설치:

```
# cp adodb330.tgz /usr/local/apache/htdocs
# cd /usr/local/apache/htdocs
# tar xvzf adodb330.tgz
```

* Acid 설치:

```
# cp acid-0.9.6b23.tar.gz /usr/local/apache/htdocs
# cd /usr/local/apache/htdocs
# tar xvzf acid-0.9.6b23.tar.gz
```

참고로 여기에서 웹 디렉토리는 /usr/local/apache/htdocs 라고 가정하였으며, ACID 에서는 /usr/local/apache/htdocs/acid 디렉토리에 있는 acid_conf.php 파일만 아래와 같이 수정해 주면된다.

```
$DBlib_path = "/usr/local/apache/htdocsadodb";
$DBtype = "mysql";
$alert_dbname = "snortdb";          // db 이름
$alert_host = "localhost";          // local 의 db 에 접근하므로 localhost
$alert_port = "3306";               // mysql 은 기본적으로 3306/tcp를 사용한다.
$alert_user = "snortuser";              // db 유저
$alert_password = "snort_pass";     // snort의 암호
$archive_dbname = "snortdb";        // db 이름, 같은 db를 사용한다.
$archive_host = "localhost";
$archive_port = "3306";
$archive_user = "snort";
$archive_password = "snort_pass ";
$ChartLib_path = "/usr/local/apache/htdocs/jpgraph-1.11/src";
$chart_file_format = "png";
```

위와 같이 설정후 snort 를 설치하도록 하자. 이번에 설치될 snort 는 mysql 과 연동하여 로그 정보를 mysql db 로 저장할 것이므로 앞에서 설치했던 방법과 다소 차이가 있다.

[그림 38] mysql 과 연동하여 snort 설치

snort 의 로그는 mysql 에 db 형태로 저장할 것이므로 configure를 실행할 때 반드시 "--with-mysql" 옵션을 추가해 주어야 한다. 그리고 snort 의 설정 파일인 snort.conf 파일에 아래의 output 설정을 추가해 IDS에서 탐지한 이벤트를 db 에 저장하도록 한다.

```
output database: log, mysql, user=snort password=snort_pass dbname=snortdb host=localhost
```

이제 snort 가 사용할 mysql db를 설정할 차례이다. 여기에서 db 이름은 snortdb 로 하고 db id와 pw 는 각각 snort / snort_pass 로 설정한다고 가정한다.

```
# mysql -u root -p mysql
mysql> SET PASSWORD FOR snort@localhost=PASSWORD('snort_pass');
mysql> create database snortdb;
mysql> grant INSERT,SELECT on snort.* to snortdb@localhost;
mysql> exit
```

다음 snort 의 소스 디렉토리(또는 schema 디렉토리)에서 아래의 명령어를 실행한다.

```
$ mysql -u snort -p snortdb < ./contrib/create_mysql
>Enter password: snort_pass
```

다음으로 snortdb-extra.gz 파일을 압축해제 한 후 역시 아래의 명령어를 실행한다.

```
$ gzip -d ./contrib/snortdb-extra.gz
$ mysql -u snort -p snortdb < ./contrib/snortdb-extra
```

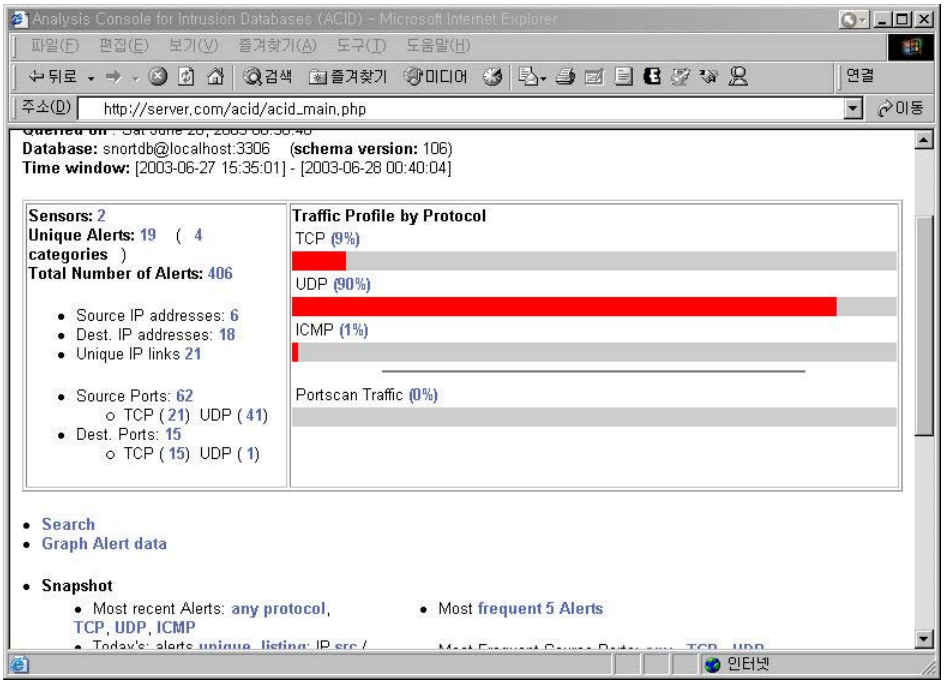
마지막으로 acid에서 제공하는 acid table을 추가하도록 한다.

```
$ mysql -u snort -p snortdb
</usr/local/apache/htdocs/acid/create_acid_tbls_mysql.sql
```

이후 db 에 로그인후 'show tables' 를 실행하면 아래와 같이 보일 것이다.

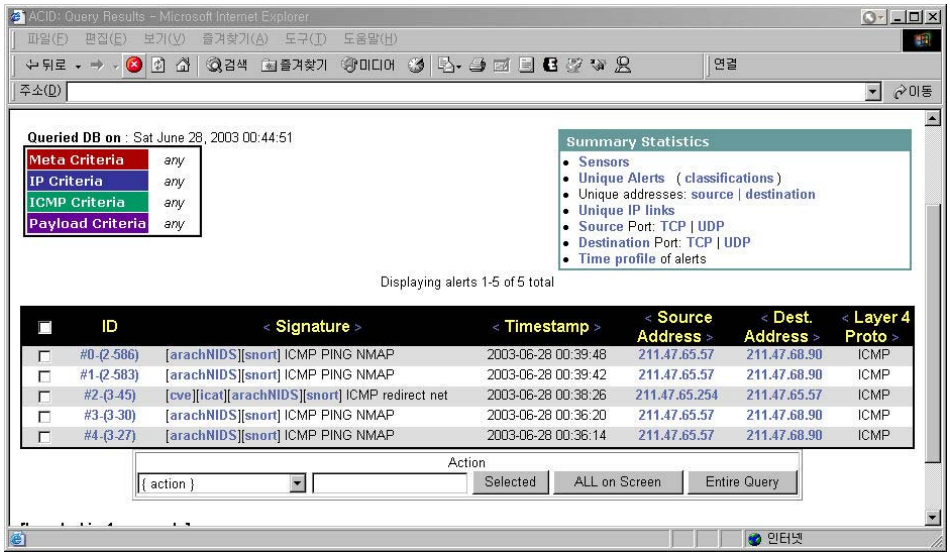
이제 모든 설정이 끝났다. mysql 과 snort를 가동하고 정상적으로 침입을 탐지하는지 확인하면 된다. 만약 트래픽이 잘 보이지 않는다면 nmap 이나 nessus 와 같은 프로그램을 이용하여 자가 스캔 또는 취약성 점검을 해 보기 바란다. 잠시 후 서버에 아래의 주소로 접속하면 다음과 같이 ACID 화면이 뜨는 것을 확인할 수 있다.
http://서버이름/acid/

```
mysql> show tables;
+------------------+
| Tables_in_snortdb |
+------------------+
| acid_ag          |
| acid_ag_alert    |
| acid_event       |
| acid_ip_cache    |
| data             |
| detail           |
| encoding         |
| event            |
| flags            |
| icmphdr          |
| iphdr            |
| opt              |
| protocols        |
| reference        |
| reference_system |
| schema           |
| sensor           |
| services         |
| sig_class        |
| sig_reference    |
| signature        |
| tcphdr           |
| udphdr           |
+------------------+
23 rows in set (0.00 sec)
>
```



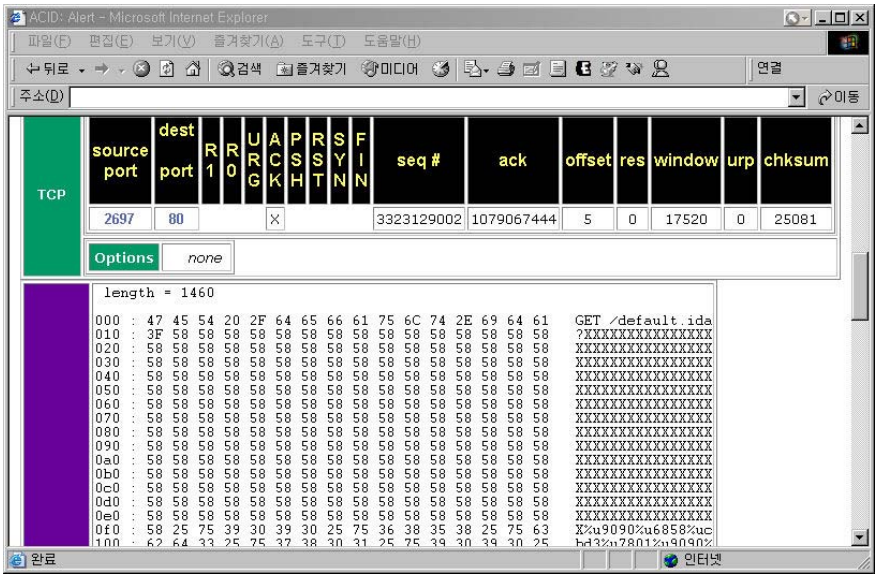
[그림 39] acid 의 메인화면

메뉴는 다소 조잡하지만 꼭 필요한 정보만 제공하고 있다. 그리고 여러 가지 메뉴를 제공하고 있으므로 각각의 메뉴를 클릭해 보기 바란다. 통계치등 여러 가지 흥미로운 정보를 확인할 수 있을 것이다.



[그림 40] acid 의 이벤트 질의결과

위 그림은 snort IDS에 생성된 로그 중 ICMP에 관련된 로그 정보만 질의한 결과이다.



[그림 41] 탐지한 패킷의 디코딩 화면

위 그림은 탐지된 패킷의 상세한 ASCII 디코딩 정보를 제공하고 있다.

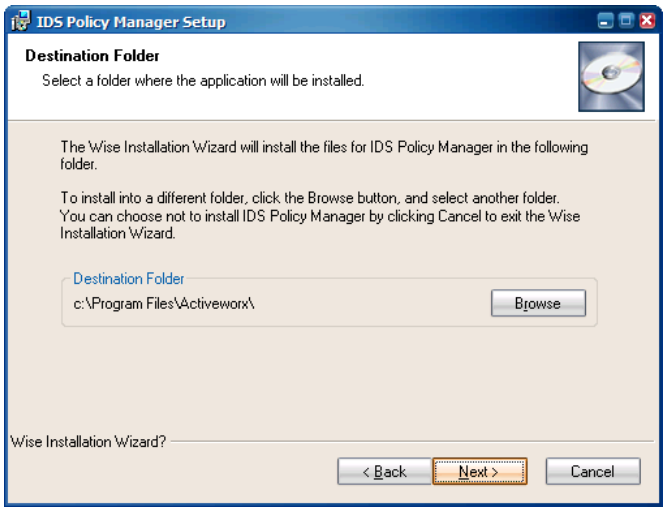
나. IDS Policy manager를 이용한 snort 관리

여러개의 snort를 운영하기 위해 개발된 프리소프트웨어인 IDS Policy Manager를 이용하여 snort를 관리하는 법을 알아보자. 다운로드는 아래의 링크에서 가능하다.
http://www.activeworx.org/downloads/idspm.v1.6.0.msi

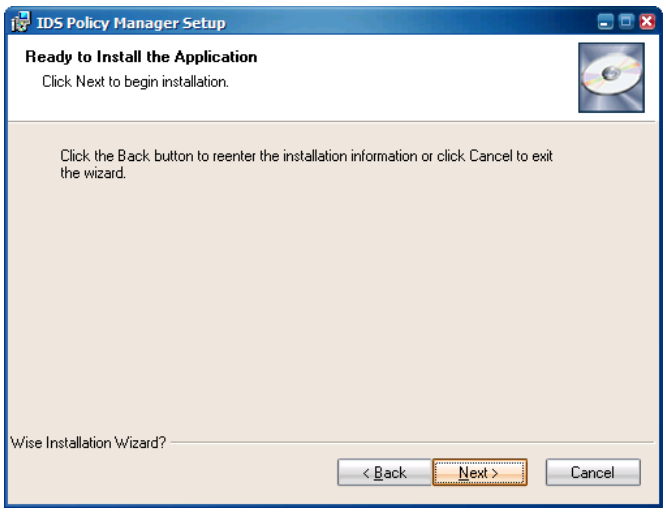
사용하는 법은 조금만 살펴보면 별로 어렵지 않을 것이니, 필요한 부분만 요약해서 정리하도록 하겠다. 그럼 설치부터 알아보도록 하자. 설치 순서는 다음과 같다.

① 설치 시작 화면
(“Next” 클릭)

② 설치 폴더 선택
(“Next” 클릭)



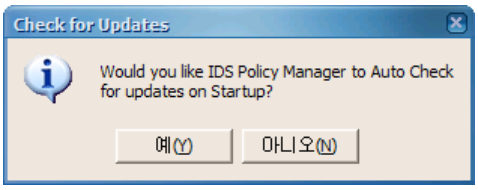
③ 설치 준비 확인
(“Next” 클릭)



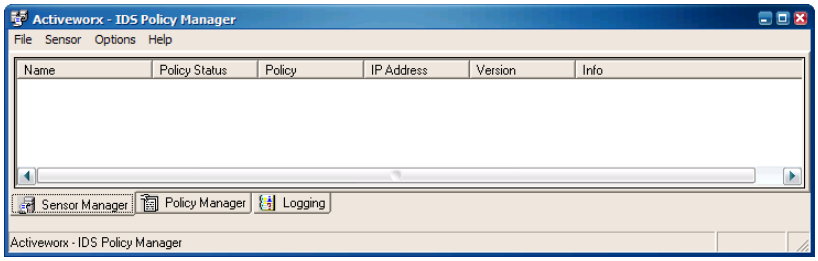
④ 설치 완료 확인 (“Finish" 클릭)

설치가 끝났으면 실행해보도록 하자. 아래와 같은 순서로 사용하면 별 무리 없을 것이다.

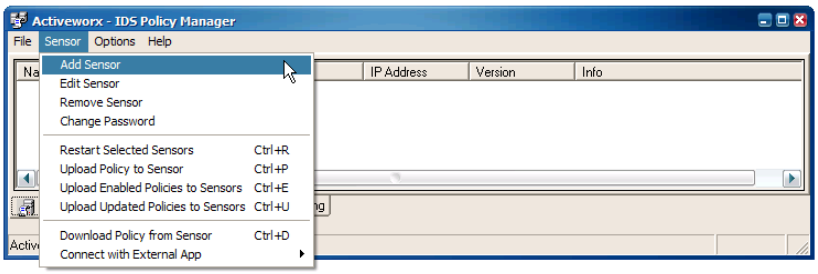
① 최소 실행시 업데이트 체크 (“예(Y)" 클릭)



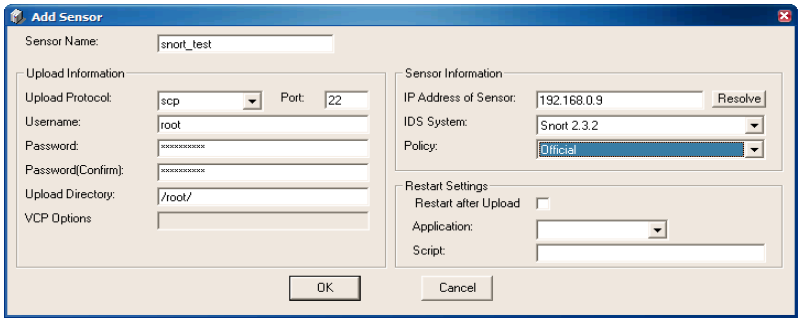
② 실행시 초기 화면



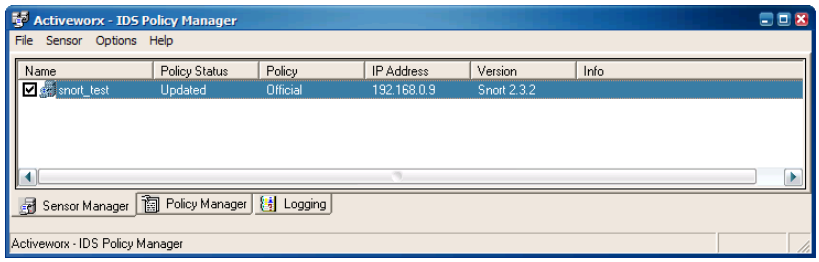
③ snort 센서 추가 (Sensor ->Add Sensor 선택)



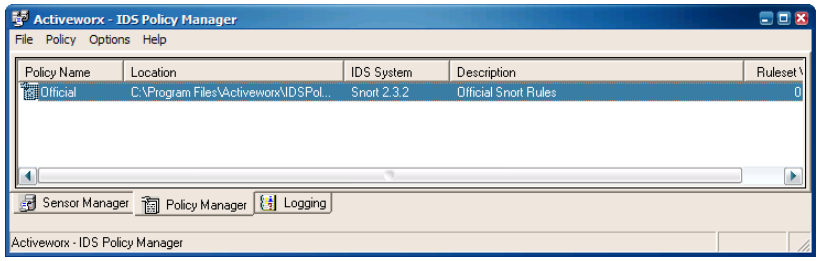
④ snort 정보 등록
(snort가 설치된 시스템 정보를 입력)



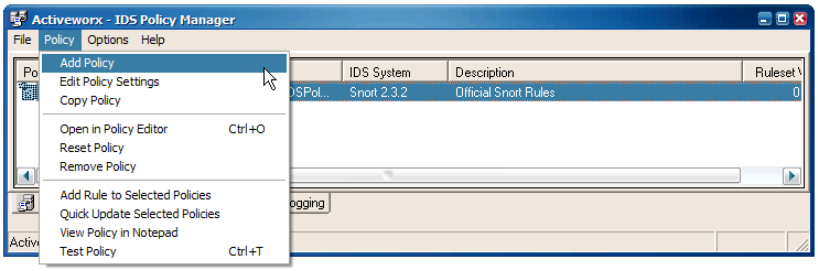
⑤ 등록 후 센서 확인



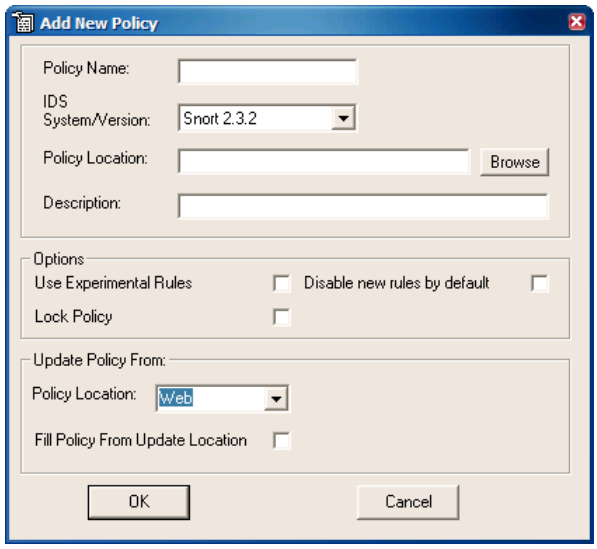
⑥ 보안정책 확인
(PolicyManager 클릭)



⑦ 보안 정책 추가
(Policy
-> Add Policy 선택)

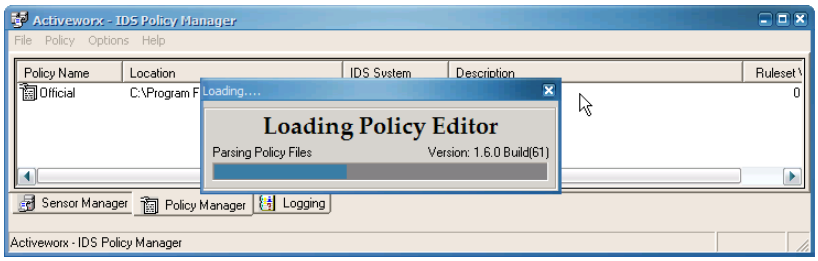


⑧ 보안 정책 등록
(각 항목에 해당 내용 입력)



* 보안정책은 기본권장

⑨ 보안 정책 확인
(PolicyName 더블클릭)



이상으로 IDS Policy Manager 사용법에 대해 간단하게 살펴보았다. 윈도우 환경에 익숙한 사용자들에게 별 무리 없이 사용이 가능하도록 되어 있으므로, 다수의 Snort를 관리하는 사용자에게는 유용하다.