HomeWork #2 연습문제

- Snort 구축 및 사용법-

학　　과　컴퓨터공학과
학　　번　201211704
이　　름　김기홍
제 출 일　2016.11.04

부경대학교

목 차

출 처 : KISA Snort를 이용한 IDS 구축(http://twoseven.kr/linux2/files/snort.pdf)
CentOS에서 Snort 설치(http://lureout.tistory.com/234)
[1]**libdnet** 설치(http://libdnet.sourceforge.net/)
[2]**libpcap** 설치(http://www.tcpdump.org/#latest-release)

---

1) libdnet : 저수준의 네트워크 방식들을 단순화 하고 휴대가능한 인터페이스를 제공한다.
2) libpcap : Packet Capture의 약자이며, 컴퓨터 네트워크 관리 분야에서 네트워크 트래픽 포착용 API를 구성하고 있다.

1. 서론

· 머리말

- 본 과제는 Snort 구축 및 간단한 테스트의 과정을 보이는 내용이다.
  필자는 보고서 작성을 위해 인터넷을 주로 활용하였고 그 과정에서 [3]**IDS** 개념 및 구축 방법을 숙지하였다. 그리고 필요한 도구 및 라이브러리에 대한 정보의 출처는 명확히 기재하였다.
  구축 환경은 Linux CentOS 64bit이며, 가상머신(Oracle VM VirtualBox)에서 구동한 가상 시스템에서 작업을 진행하였다.
  이 보고서는 Snort를 직접 구축하는 데 있어서 어려움이 없도록 설치 방법을 기록하는 데 초점을 두었으며, 사용법은 여러 가지 규칙을 적용하면 되기 때문에 가장 기본적인 사용만을 보이도록 한다.

· IDS란?

- 침입 탐지 시스템이라 불리며, 방화벽(firewall)과 함께 주목받는 보안 솔루션이다. 방화벽이 IP나 포트를 기준으로 비정상 트래픽을 차단하는 것이라면 IDS는 포트에 대한 정보뿐만 아니라 패킷의 데이터까지 분석하여 실제로 인터넷을 통해 어떠한 위협이 발생하고 있는지를 분석할 수 있게 된다.
  하지만 IDS가 모든 공격을 정확하게 인식하여 탐지하는 것은 아니다. 몇 가지 근본적인 한계를 가지고 있는데 그 중에 가장 큰 문제는 '오탐율'이다. IDS에서 오탐은 크게 두 부류로 나눠지는데, 실제로 비정상 트래픽이지만 이를 탐지 하지 못하는 경우를 'False Negative'라 하며, 비정상 트래픽이 아닌 정상적인 트래픽을 탐지하는 경우를 'False Positive'라 한다. IDS를 운영하다 보면 위와 같은 오탐율이 매우 높다는 것을 알 수 있다. 따라서 이 오탐율을 줄이는 것이 IDS의 관건이라 할 수 있다. 또한 침입에 대한 탐지는 할 수 있어도 공격에 대한 대응은 부족하다는 한계가 있다.

· Snort란?

- IDS가 침입을 탐지하는 방식에는 여러 가지가 있지만 그 중에 가장 대표적인 방식은 룰(Rule) 기반의 패턴 매칭 방식으로 이는 사전에 정의된 패턴 또는 룰에 따라 트래픽이 매칭 되었을 경우 공격 또는 비정상 트래픽으로 판단하는 방식이다.
  그리고 이러한 룰 기반의 공개 IDS 프로그램 중 가장 대중적으로 사용되고 있는 프로그램이 바로 Snort 프로그램이다.

---

3) IDS : Intrusion Detection System의 약자로 일반적으로 시스템에 대한 원치 않는 조작을 탐지한다.

Snort는 실시간 트래픽 분석, 프로토콜 분석, 내용검색/매칭, 침입탐지 Rule에 의거하여 오버플로우, 포트스캔, CGI공격, OS확인 시도 등의 다양한 공격과 스캔을 탐지할 수 있다. 그리고 침입탐지 Rule은 보안 커뮤니티를 통해 지속적으로 업데이트 되고 또한 사용자가 직접 Rule을 작성하여 추가할 수 있도록 설계되어 있어 최신공격에 대한 적응에 빨리 대처할 수 있다.

· Snort 구축에 필요한 도구 및 라이브러리

- Snort를 구축하기 전에는 사전에 설치해야 할 항목이 몇 가지 있다. 먼저 기본적인 Snort 구축을 위해서는 크게 'Development Tools', 'Development Libraries', [4]**pcre**, **pcre-devel**, [5]**flex**, **bison**, [6]**zlib**, **zlib-devel**, libdnet, libpcap 등이 있다.
  그리고 Snort 공식 홈페이지(http://www.snort.org)에서 제공하는 snort 파일과 [7]daq 파일이 필요하다.

2. 본론

· Snort 구축을 위한 사전 준비

- 서론에서 언급했던 도구 및 라이브러리를 설치하기 위해서 yum 명령어를 사용했다. 예를 들어 'Development Tools'나 'Development Libaries'와 같은 것은 yum -y groupinstall "~" 명령을 사용했으며, 나머지(libdnet, libpcap 제외)는 yum -y install ~ 명령으로 설치가 가능하다.
  그 후에는 yum update ; yum upgrade 명령으로 현재 설치되어 있는 소프트웨어들을 최신 상태로 유지시킨다.
  그리고 libdnet, libcap는 구글(http://www.google.com)에 검색하여 다운로드 페이지 링크를 얻어 설치 할 수 있다.

---

4) pcre, pcre-devel : 트래픽을 처리하는 과정에서 패킷을 모니터링하며 실시간으로 트래픽을 분석하고 IP네트워크 상에서 침입탐지의 패턴을 분석하는 일종의 패턴룰이다.
5) flex, bison : flex는 String으로 적혀 있는 어떤 컴퓨터의 언어의 어휘 분석을 하여 의미 단위인 토큰을 추출하고 bison은 그 토큰들의 관계가 어떻게 되는지를 따져서 실제로 그런 관계가 되도록 한다.
6) zlib, zlib-devel : C로 작성된 데이터 압축 라이브러리의 일종이다.
7) daq : Data Acquisiton의 약자이며, 패킷의 입출력을 위한 API 라이브러리이다.

http://libdnet.sourceforge.net/(위)
http://www.tcpdump.org/#latest-release(아래)

필자는 위의 설치된 파일들을 모두 /root/snort 디렉토리에 모아 /usr/local 디렉토리에 압축을 풀어서 생긴 디렉토리에서 ./configure ; make ; make install 명령으로 설치를 마친다.

/root/snort 디렉토리 및 각 파일 설치 명령어

· Snort 설치

- 사전 준비를 마쳤다면 큰 어려움 없이 Snort 설치가 가능하다. Snort도 유사하게 압축을 푼 후 해당 디렉토리에서 ./configure ; make ; make install 명령을 통해 설치할 수 있다. (단, 필요한 도구 및 라이브러리가 없을 경우에는 ERROR가 발생할 수 있다. ERROR 발생 시 내용을 확인하여 필요한 파일을 먼저 설치해주어야 Snort 설치가 가능하다.)

· Snort Rules 설정

- 이제 Snort 설치를 마쳤으니 Snort 공식 홈페이지에서 최신 Rule을 다운로드 해야 한다. 다운로드 후에 설정하는 방법은 명령어로 대체한다.

```
# mkdir /etc/snort
# mkdir /var/log/snort
# cd /root/snort
# tar zxf snortrules-snapshot-2962.tar.gz -C /usr/local
# cd /usr/local
# cp etc/* /etc/snort
# touch /var/log/snort/alert
# chmod 600 /var/log/snort/alert
# mkdir /usr/local/lib/snort_dynamicrules
# cp -r /usr/local/so_rules /etc/snort
# cp /etc/snort/so_rules/precompiled/Centos-5-4/i386/2.9.6.2/*.so ₩
/usr/local/lib/snort_dynamicrules
# cp -r /usr/local/rules /etc/snort
# cat /etc/snort/so_rules/*.rules >>/etc/snort/rules/so-rules.rules
# ln -s /etc/snort/rules/so-rules.rules /etc/snort/rules/so_rules.rules
```

3. 결론

· Snort 구축 결과

- 간단하게 Snort의 설치 여부를 확인하기 위해 터미널에서 snort -V 명령을 입력해 보면 아래와 같이 Snort는 잘 설치되었음을 알 수 있다.

```
[root@localhost Desktop]# snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.6 GRE (Build 285)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.32 2012-11-30
           Using ZLIB version: 1.2.7
```

snort -V 명령어 입력

snort 의 옵션은 아래 표를 참고하자.

옵션	기능
-A	alert 모드를 fast, full, none 중의 하나로 지정
-a	ARP 패킷을 프린트
-b	패킷을 tcpdump 파일로 저장. 바이너리 포맷이므로 저장 속도가 빨라짐.
-c [file]	[file]로 지정된 파일을 룰 파일로 사용함.
-C	패킷의 사용자 데이터 부분을 문자만 프린트
-D	snort를 데몬 모드로 실행. 바로 백그라운드로 들어가고 터미널을 종료해도 계속 돌게 됨.
-F	BPF 필터링식으로 지정된 파일에서 읽어옴.
-g	snort의 gid를 그룹으로 설정.
-h	홈네트워크 변수 HOME_NET을 세팅.
-i	지정된 네트워크 인터페이스를 모니터링. 값으로 eth0, eth1 등이 올 수 있음.
-I	alert 결과물에 네트워크 인터페이스 이름을 붙임.
-l [dir]	지정된 디렉토리에 로그 데이터를 저장.
-n	개인 패킷만을 모니터링하고 프로그램을 종료.
-N	로깅 기능을 사용하지 않고 alert만이 저장.
-o	룰셋 테스트 순서를 Pass, Alert, Log순서로 바꿈.
-O	IP주소를 알 수 없도록 표시.
-p	무작위 모드를 사용하지 않고 스니핑을 함.
-q	패킷의 [8]**snaplen**을 지정. default는 1514
-r	지정된 tcpdump 파일의 패킷들에 대해서 IDS 엔진을 돌림.
-s	alert 로그 메세지를 syslog 시스템을 통해 시스템에 보냄.
-S	룰 파일의 var로 지정된 변수를 재정의 할 수 있음.
-t	초기화 후 디렉토리로 chroot함.
-u	초기화 후 snort의 uid를 사용자로 바꿈.
-v	많은 메시지를 뿌림.
-V	버전 정보를 표시.
-X	링크 레이어의 로우 패킷 데이터를 덤프.
-e	두번째 레이어의 헤더 정보를 프린트.
-d	어플리케이션 레이어를 덤프.
-?	도움말을 보여줌.

snort 실행 옵션

· 실제 Snort 테스트(ICMP 탐지)

- 이제 설치한 Snort가 정상적으로 탐지를 하는지 테스트 해보자. 먼저 ICMP(Ping)을 탐지하는 Rule을 추가하기 위해 /etc/snort/rules/local.rules와 /etc/snort/sid-msg.map파일에 vi에디터를 통해 내용을 추가해주어야 한다.

  vi /etc/snort/rules/local.rules 명령으로 에디터를 실행시킨 다음 맨 마지막 행에 alert icmp any any -> any any (msg:"ICMP Test"; sid:100001;)을 추가해주고, vi /etc/snort/sid-msg.map 명령을 한 다음 맨 마지막 행에 100001 || ICMP Test를 추가해준다.

```
[root@localhost Desktop]# vi /etc/snort/rules/local.rules

alert icmp any any -> any any (msg:"ICMP Test"; sid:100001;)


vi /etc/snort/sid-msg.map

100001 || ICMP Test
```

ICMP 탐지 Rule 추가

이제 snort -c /etc/snort/rules/local.rules 명령을 통해 snort를 실행해보자. 그리고 로그를 확인하기 위해 터미널을 하나 새로 연 다음 tail -f /var/log/snort/alert을 입력하고 대기한다.

```
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/rules/local.rules"
Tagged Packet Limit: 256
Log directory = /var/log/snort

[root@localhost snort]# tail -f /var/log/snort/alert
```

snort 프로그램 실행 및 로그 기록 출력 대기 모습

8) snaplen : 캡처할 수 있는 패킷의 최대 크기.

이제 snort로 Ping을 보내보면 아래와 같이 로그가 올라오는 걸 볼 수 있다.
한 번의 ping 전달이 ECHO와 ECHO REPLY 두 가지 로그로 기록되는 것은 좀 전의 설정에서 any any -> any any에서 모두 알림하게 설정을 했기 때문이다.

```
[root@localhost snort]# tail -f /var/log/snort/alert
11/22-14:27:27.880238 10.0.2.15 -> 203.250.123.225
ICMP TTL:64 TOS:0x0 ID:56449 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:14241   Seq:4  ECHO

[**] [1:100001:0] ICMP Test [**]
[Priority: 0]
11/22-14:27:27.882718 203.250.123.225 -> 10.0.2.15
ICMP TTL:127 TOS:0x0 ID:312 IpLen:20 DgmLen:84
Type:0  Code:0  ID:14241  Seq:4  ECHO REPLY
```

log 기록 실시간 출력

이로써 간단하게 snort가 제대로 작동하는지에 대한 테스트가 끝이 났다. Snort의 Rule 설정은 정말 다양하게 설정할 수 있다. (다른 다양한 Rule 설정에 관해서는 (http://sinun.tistory.com/152) 블로그에 Snort Rule Content 키워드 정리가 잘 정리되어 있다.)

· 구축 소감

- Snort를 구축하기 위한 사전도구나 라이브러리에 대해 조사를 해봄으로써 설치만 하는 것도 만만치 않다는 것을 느꼈다. 또한 다양한 규칙들을 활용하는 것도 생각보다 쉽지 않았다. 하지만 실제로 직접 구축하고, 간단하게 사용해보는 과정에서 여러 네트워크 공격법과 IDS에 대한 이해를 증진시킬 수 있었다. 비록 설치 및 간단한 사용밖에 해보지 못했지만 보안 수업을 수강하는 입장에서 새로운 것을 하나 더 배운 것 같아서 좋았다.