시스템 및 네트워크 모니터링을 통한 보안 강화

오늘과내일 넷센터 홍석범 (antihong@tt.co.kr)

1. 포트(Port)의 개념과 포트 제어의 모든 것
2. 내부 시스템(호스트) 모니터링 소프트웨어 활용
3. 외부 시스템(네트워크) 모니터링 소프트웨어 활용

1. 포트의 개념과 포트 제어의 모든 것

시스템의 접속은 포트와 포트간의 통신이라고 할 정도로 포트의 의미는 매우 중요하다. 당연히 시스템의 모니터링을 위해서는 포트의 개념에 대한 이해가 매우 중요한데, 단순히 다음 파트에서 설명할 시스템 및 네트워크 모니터링 프로그램에 대한 사용 방법을 익히는 것도 좋지만 사전에 포트에 대한 기본적인 개념을 이해하고 있어야 각종 현상에 대한 이해 및 상황에 대한 즉각적인 대처가 가능하고, 또한 문제 발생시 원활히 처리를 할 수 있다.
이번 호에서는 서버를 운영하면서 자주 다루어지는 포트(Port)의 개념과 포트 제어 방법에 대해 알아보도록 하겠다.

포트의 개념과 포트 분류의 원칙.

시스템간에 통신을 할 때 물리적인 전송선은 하나이지만 그것을 여러 개의 응용 프로그램들이 서로 나누어 사용하기 위해서 포트(Port) 라는 개념이 도입되었다. 한 시스템 내의 소켓(Socket) 을 사용하는 모든 프로세스는 별도의 포트 번호가 할당된 소켓을 갖게 되는데, 이것은 TCP/IP 가 지원하는 상위 계층의 응용 프로그램을 구분하기 위한 번호이다.
서버와 클라이언트간의 모든 네트워크는 일정화 된 규칙에 따라 포트를 이용해 통신을 하게 된다. 일반적으로 포트 번호로 16 비트를 사용하며 사용 가능한 포트는 0 번부터 65535 번까지인데 이 중 특히 0 번부터 1023 번까지를 특별히 Privileged Port 라 하고 1024 번부터 65535 번 포트까지를 Unprivileged Port 라고 한다. 포트 바인딩등과 같은 소켓 프로그램을 이용하거나 80 번을 사용하는 아파치 웹서버와 같이 데몬을 셋팅하여 특정한 포트를 점유하는 프로그램을 실행할 수 있는데, Privileged Port 영역내에 있는 포트를 이용하는 프로그램을 작동하려면 반드시 root 권한으로 작동하여야 한다. 즉, 다른 말로 표현하면 root 이외의 유저라면 1024 번부터 65535 번까지의 포트만을 생성(바인딩)할 수 있는 것이다. 물론 root 권한으로는 0 부터 1023 사이의 포트뿐만이 아니라 1024 번 포트 이후의 포트도 사용 가능하다.

그러나 root 권한으로 작동하는 대부분의 데몬들은 0 번부터 1023 번 사이에 존재하고 1024 번 이후의 포트는 주로 일반 유저 권한으로 데몬을 실행하거나 클라이언트가 서버와 통신시 사용된다. 따라서 1024 이후의 포트를 흔히 클라이언트 포트라고 부르기도 한다.
참고로 현재의 시스템에서 클라이언트가 사용 가능한 포트는
sysctl -a | grep local_port 로 확인 가능하며
sysctl -w net.ipv4.ip_local_port_range="32768 61000" 와 같은 명령어로 설정 변경이 가능하다. 클라이언트의 포트는 OSI 7 Layer 에서 7 계층인 Application Layer 에서 결정되게 되는데, 예를 들어 웹 브라우저를 이용해 http://www.tt.co.kr/ 이라는 웹 서버를 접속하였다면 서버측에서는 www 포트인 80 번 포트가 반응하게 되고, 클라이언트에서는 웹 브라우저에서 1024 번 이후의 임의의 포트를 할당하여 서로 통신이 되는 것이다.
PC 에서 telnet 이나 ftp 접속을 한 후 netstat –na 를 실행하여 클라이언트 PC 에서 반응한 포트를 보면 1024 이후의 포트에서 임의로 할당되는 것을 확인할 수 있을 것이다.
이 외에 Well Known Port 라는 것이 있는데, 이는 IANA 에서 TCP 와 UDP 에 대해 정책적으로 할당한 포트로서 www 는 80 번, SMTP 는 25 번, telnet 은 23 번 등과 같이 통상적으로 약속한 프로토콜이 사용하는 포트에 대한 정의인데, 이는 /etc/services 파일에 규정되어 있다.
이 원리를 이용하여 아래의 문제를 풀어보도록 하자.
갑자기 서버에 과부하가 걸리고 있는 듯 하여 ps aux 로 시스템의 프로세스 상황을 살펴 보니 아래와 같이 sendmail 프로세스가 많이 떠 있는 것을 확인하였다.
그런데, 아시다시피 sendmail 은 외부에서 서버로 보내어진 메일을 받는 역할(Local 은 25 번 포트 작동, Foreign 은 Unprivileged Port 작동) 과 내부에서 큐로 보내어진 메일을 외부로 발송하는 기능(Local 은 Unpriviliged port 작동, Foreign 은 25 번 포트 작동)이 있는데, 아래의 sendmail 프로세스는 이 두 가지 기능 중 전자와 같이 외부에서 내부로 보내어지고 있는 메일을 받는 프로세스일까? 아니면 후자와 같이 현재 서버에서 외부로 발송되는 메일을 처리하는 프로세스일까?

[root@www /root]**# ps aux|grep sendmail**

root	3425	0.0	0.2	2472	1256 ?	S	Aug10	0:10 sendmail: accepti
root	20818	0.0	0.2	2596	1520 ?	S	01:07	0:00 sendmail: server
root	24572	0.0	0.3	2728	1684 ?	S	01:32	0:00 sendmail: sevrer
root	24970	0.0	0.2	2596	1524 ?	S	01:34	0:00 sendmail: server
root	25811	0.0	0.2	2596	1524 ?	S	01:38	0:00 sendmail: server
root	13455	0.0	0.2	2472	1256 ?	S	Aug11	0:10 sendmail: accepti
root	23766	0.0	0.2	2344	1520 ?	S	01:07	0:00 sendmail: server
root	47932	0.0	0.2	2432	1524 ?	S	01:34	0:00 sendmail: server
root	98793	0.0	0.2	2342	1524 ?	S	01:38	0:00 sendmail: server

이 때의 해결책은
위 결과만으로는 알 수 없으며 아래와 같이 netstat 을 이용하면 된다.
Sendmail 이 작동한다면 Local 이든 Foreign 이든 하나는 반드시 25 번 포트가 사용된다는 특징을 이용하면 된다.

[root@www /root]# **netstat -na|grep :25**

tcp	0	0 0.0.0.0:25	0.0.0.0:*	LISTEN
tcp	0	0 211.47.65.56:25	211.47.66.71:3420	TIME_WAIT
tcp	0	0 211.47.65.56:25	198.6.49.12:36877	ESTABLISHED
tcp	0	0 211.47.65.56:25	211.119.130.90:3626	ESTABLISHED
tcp	0	0 211.47.65.56:25	210.121.167.112:3565	ESTABLISHED
tcp	0	0 211.47.65.56:25	210.111.91.130:4858	ESTABLISHED
tcp	0	0 211.47.65.56:25	128.134.24.193:4931	ESTABLISHED
tcp	0	0 211.47.65.56:25	211.54.29.139:1631	ESTABLISHED
tcp	0	0 211.47.65.56:25	203.25.16.2:2666	ESTABLISHED
tcp	0	0 211.47.65.56:25	211.54.29.139:1630	ESTABLISHED
tcp	0	0 211.47.65.56:25	192.198.165.17:51757	ESTABLISHED
tcp	0	0 211.47.65.56:25	210.219.250.207:2300	TIME_WAIT
tcp	0	0 211.47.65.56:25	210.106.207.100:1036	ESTABLISHED
tcp	0	0 211.47.65.56:25	134.239.84.2:4499	ESTABLISHED
tcp	0	0 211.47.65.56:25	202.119.208.10:42934	ESTABLISHED

이 결과를 보면 왼쪽에 보이는 것이 (Local)서버쪽의 IP:PORT 상태이며 오른쪽에 보이는 것이 (Foreign)원격지의 IP:PORT 상태이다. 위 상태를 보면 좌측의 서버(Local)측에서는 25 번 포트가 반응하고 있고, 우측의 원격지(Foreign)에서는 1024 이후의 포트중 임의의 포트 즉 클라이언트 포트가 반응하고 있는 것을 확인할 수 있다.
이를 통해 현재의 상태는 외부(원격지) 에서 로컬 서버로 메일을 전송하고 있는 상태(즉 서버가 외부에서 발송된 메일을 받고 있는 상태)라는 것을 알 수 있다.
그리고 만약 아래와 같이 반대의 상황이라면 앞의 상황과는 반대로 외부의 원격지에서 25 번 포트가 반응하고 있으므로 내부 로컬에서 외부로 메일을 발송하고 있는 상태라는 것을 알 수 있다.

[root@www /root]# **netstat -na|grep :25**

tcp	0	0 0.0.0.0:25	0.0.0.0:*	LISTEN
tcp	0	0 211.47.65.56:38799	211.47.66.71:25	TIME_WAIT
tcp	0	0 211.47.65.56:43634	198.6.49.12:25	ESTABLISHED

tcp	0	0	211.47.65.56:43891	211.119.130.90:25	ESTABLISHED
tcp	0	0	211.47.65.56:7478	210.121.167.112:25	ESTABLISHED
tcp	0	0	211.47.65.56:1234	210.111.91.130:25	ESTABLISHED
tcp	0	0	211.47.65.56:2152	128.134.24.193:25	ESTABLISHED
tcp	0	0	211.47.65.56:3645	211.54.29.139:25	ESTABLISHED
tcp	0	0	211.47.65.56:3601	203.25.16.2:25	ESTABLISHED

포트 번호외에 프로토콜에 따라 TCP 와 UDP 포트로도 분류할 수 있는데 telnet, sendmail, httpd 등 대부분의 프로토콜은 신뢰성이 보장되는 TCP 를 사용하나 DNS 등 특정 프로토콜의 경우 UDP 를 사용하거나 TCP 와 UDP 를 함께 사용하는 경우도 있다. UDP 패킷은 TCP 패킷에 비해 헤더에 패킷의 순서에 대한 정보가 없기 때문에 패킷의 신뢰성 보다는 오버 헤드가 없고 빠른 속도를 필요로 할 때 쓰이며 한 시스템 이상에게 메시지를 전달하는 멀티 캐스트로 사용되기도 한다.

포트 점검 및 제어

그렇다면 내가 운영하는 서버에는 어떤 포트가 떠서 서비스 요청을 기다리고 있는지 알 수 있는 방법이 있을까? 간단히 검색하고자 하는 서버에 대해 포트 스캔을 해 보면 된다. 특정 포트가 반응하는지 여부는 여러 방법으로 확인 가능한데, 단순히 특정 포트로 telnet 접속하여(예: telnet www.server.com 522) 접속이 되는지 여부를 확인하는 방법도 있고, 별도의 포트 스캔 전용 프로그램을 사용해도 된다. 포트 스캔을 위해 가장 권장할 만한 프로그램으로는 nmap 이 있는데, http://www.insecure.org/nmap/ 에서 tarball 형태의 소스를 다운로드 받아 설치하거나 http://rpmfind.net/ 에서 rpm 형태의 파일을 검색한 후 다운로드하여 사용하여도 된다.

nmap – p 1–65535 localhost 를 하면 현재 자신의 시스템에 어떤 포트가 떠 있는지 모든 포트에 대해 확인 가능하다. 만약 아무런 옵션을 주지 않고 nmap localhost 만 실행하면 /etc/services 파일에 정의된 Well Known Port 에 대해서만 스캔을 하게 되는 것을 주의하기 바란다.

또한 nmap -p 21,23,53,80 192.168.1.0/24 와 같이 스캔할 경우에는 192.168.1.0 대역(즉 192.168.1.1 부터 192.168.1.254 까지 모든 호스트) 에 대해 21,23,53,80 번 등의 특정 포트가 열려있는지에 대해서만 스캔을 하게 된다.

그런데, 스캔을 한 결과를 보니 자신이 알 지 못하는 이상한 포트가 떠 있는 경우가 있다. 일반적으로 해킹을 한 후에는 이후에 원격에서도 쉽게 접속할 수 있도록 특정한 포트에 백도어를 설치하여 숨겨놓는 경우가 있으므로 백도어가 아닐까 의심하는 경우가 있는데, 이 포트를 사용하는 프로세스가 어떤 것인지 확인하는 방법과 이 포트를 죽이는 방법은 무엇일까?

.

Port	State	Service
21/tcp	open	ftp
23/tcp	open	telnet
25/tcp	open	smtp
80/tcp	open	http
110/tcp	open	pop-3
1234/tcp	open	hotline
3306/tcp	open	mysql
4321/tcp	open	rwhois
7755/tcp	open	unknown
10101/tcp	open	unknown

만약 포트 스캔 결과 위와 같이 나왔을 경우 우측에 telnet, ftp 등 포트번호에 해당하는 서비스의 이름이 나오는데, 이는 단순히 스캔된 포트에 대하여 /etc/services 파일에 정의된 서비스명과 매치를 시켜 놓은 것 뿐이며 80 번 포트라고 해서 반드시 웹데몬이 아닐 수도 있음을 주의하기 바란다. 아울러 unknown 이라고 나온 것은 /etc/services 파일에 정의되지 않은 포트이기 때문이다. 위 스캔 결과에서 10101 번 포트가 수상하여 10101 번 포트를 쓰고 있는 프로세스가 무엇인지 알고 싶은 경우 알 수 있는 방법은 아래와 같이 두 가지가 있다.

첫번째는 lsof 를 이용하는 방법이고, 두번째는 fuser 라는 명령어를 이용하는 방법이다. 이 두 명령어는 대부분의 시스템에서 기본적으로 이용 가능한데, 혹 명령어가 실행되지 않으면 해당 패키지를 찾아 설치하면 된다. Lsof 는 lsof 라는 패키지에, fuser 는 psmisc 패키지에 포함되어 있다. 이 패키지는 http://rpmfind.net/ 에서 rpm 으로 다운로드 가능하다.

1) lsof 를 이용시

lsof 는 LiSt Open Files 의 약자로서 현재의 시스템에서 프로세스가 오픈하고 있는 모든 파일과 상세 정보를 보여주는데 **lsof –i | grep 10101** 을 하면 10101 번 포트를 사용하고 있는 프로세스를 확인할 수 있다. 만약 단순히 lsof 실행 파일만 다른 시스템에서 복사하여 사용하는 경우에는 –i 옵션이 제대로 작동하지 않는 수가 있는데, 이러한 경우에는 lsof > result.txt 를 실행하여 result.txt 파일의 결과를 참고해도 된다. 이 외에도 lsof 는 많은 다른 목적으로 사용되는 유용한 프로그램이므로 사용 방법을 익혀 두는 것이 좋다.

2) fuser 를 이용시

fuser 는 특정한 파일이나 파일 시스템을 사용하고 있는 프로세스의 PID 를 보여주는 명령어로 **fuser –n tcp 10101** 과 같이 입력하면 10101 번 포트를 사용중인 프로세스 ID 를 보여준다. 물론 UDP 포트일 경우에는 fuser –n udp 53 과 같이 사용하면 된다.

이때 프로세스 ID 가 570 번이라는 것을 확인했다면 PID 570 번을 사용하는 프로세스를 확인하여야 하는데, 이는 ps 에서 보이는 결과로 확인을 하거나 **ls –la /proc/570/** 의 결과중 exe→ 가 가리키는 결과를 확인 하면 된다. exe→ 가 가리키는 경로는 현재 실행 파일을 의미하며 cmd→ 는 실행 파일이 참조하는 파일이나 디렉토리의 경로를 알려준다.
만약 exe → /home/user1/hacking/hacking* 이라고 되어 있으면 이 경로에 있는 파일이 10101 번 포트를 바인딩하고 있는 프로세스라는 것을 확인할 수 있다.

그리고 만약 이 프로세스가 정상적인 것이 아니라면 kill –9 570 (570 은 PID) 또는

killall –9 –ev hacking (hacking 은 프로세스 이름) 으로 해당 프로세스를 강제로(-9) 죽이면 포트도 닫히게 된다. 또는 커널 레벨에서 작동하는 패킷 필터링 툴인 ipchains(커널 2.2.X 기반) 나 iptables(커널 2.4.X 기반) 를 이용하여 소스나 목적지 포트를 제어할 수도 있다.

들어오는 패킷에 대해서는 INPUT 을, 나가는 패킷에 대한 제어는 OUTPUT 을 그리고 소스 포트에 대한 제어는 –sport, 목적지 포트에 대해서는 –dport 를 이용하면 된다. 이를테면

```
# iptables –A INPUT –p tcp ! –sport 0:1023 –dport 25 –j ACCEPT
```

와 같은 경우 들어오는(INPUT) 패킷중 소스 포트가 0 부터 1023 이 아닌(! –sport 0-1023) 즉 1024 부터 65535 까지의 포트 그리고 목적지 포트는 25 번인 패킷(--dport 25)을 허용하겠다는(-j ACCEPT) 의미 이다.
iptables 를 이용한 패킷 필터링에 대해서는 iptables 관련 문서나 글을 참고하기 바란다.

포트가 닫혔는지 여부는 “telnet localhost port 번호” 로 확인하면 된다.
이외 netstat –lnp 와 같이 확인시에는 현재의 시스템에서 리슨(Listen)하고 있는 포트와 해당하는 프로그램의 목록을 쉽게 볼 수 있다.
또한 아래의 사이트를 참조하면 특정한 포트 번호가 어떤 역할을 하는지에 대해서 상세한 정보를 얻을 수 있으니 특정 포트의 역할에 대해 궁금하신 분은 참고하기 바란다.

http://tantalo.net/ports/index.php3
http://advice.networkice.com/advice/Exploits/Ports/default.htm
http://www.chebucto.ns.ca/~rakerman/port-table.html
http://www.networksorcery.com/enp/nav0204.htm
http://www.practicallynetworked.com/sharing/app_port_list.htm
http://www.technotronic.com/tcpudp.html

2. 내부 시스템 모니터링 소프트웨어 활용

시스템에 문제가 발생했을 때 또는 발생 가능한 사고를 사전에 예방하기 위해서는 여러 가지 방법으로 시스템을 모니터링하고 또한 문제의 원인을 파악하여 문제를 해결하여야 하는 것이 시스템 관리자의 역할이다. 이번 파트에서는 간단한 명령어를 이용하여 호스트 시스템을 모니터링 할 수 있는 방법부터 각종 응응 프로그램을 이용하여 효율적으로 시스템을 모니터링하는 방법에 대해 알아보도록 하겠다.

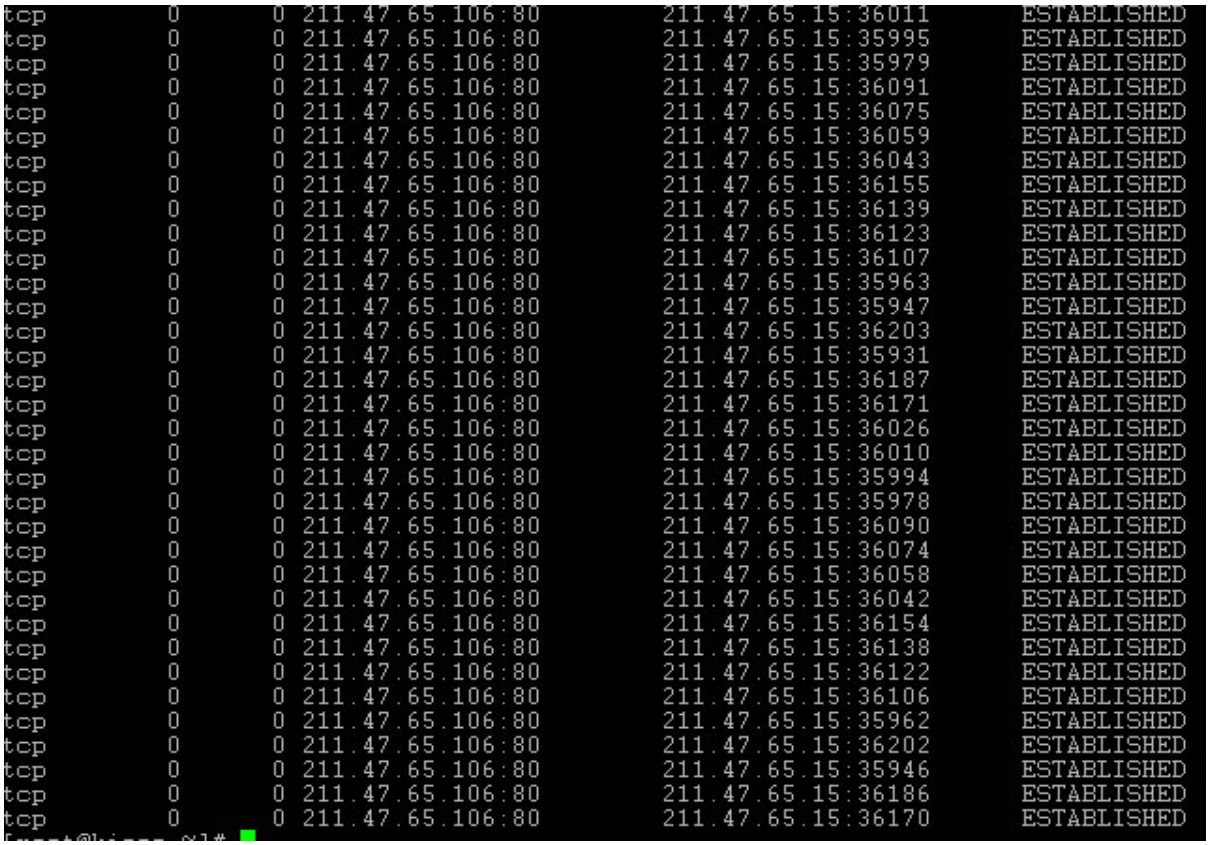
간단한 명령어로 시스템 모니터링하기.

시스템 모니터링을 위해 가장 막강하면서도 간단한 프로그램으로는 앞에서 잠깐 설명한 netstat 이 있다. netstat 은 네트워크 연결정보, 라우팅 테이블, 인터페이스 정보등을 제공하며 그림과 같이 netstat –l 을 입력시 현재 시스템에서 리슨(Listen) 하고 있는 프로그램 및 포트 정보를 알려준다.

```
[root@work ~]#
[root@work ~]#
[root@work ~]# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:www                   *:*                     LISTEN
tcp        0      0 *:ftp                   *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 *:telnet                *:*                     LISTEN
tcp        0      0 *:1241                  *:*                     LISTEN
tcp        0      0 *:3001                  *:*                     LISTEN
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node Path
[root@work ~]#
[root@work ~]#
[root@work ~]#
```

< 그림 1 netstat –l 실행 결과>

또한 SYN 패킷 요청만 보내는 서비스 거부 공격의 일종인 TCP SYN Flooding 공격(본지 7 월호 SysAdmin 내 “TCP SYN Flooding 공격의 원인과 해결책” 참고) 이나 아래와 같이 특정 데몬의 접속 요청을 독점해 버리는 Connect Flooding 공격도 감지 (netstat –na|grep EST) 할 수 있다.



< 그림 2. Connect Flood 공격 감지 화면>

이 외 netstat –s 로 각종 프로토콜에 대한 통계를 볼 수 있으며 netstat –r 로 라우팅 상황(route 명령어와 동일) 및 netstat –i 로 인터페이스 상태를 점검(ifconfig 와 동일)할 수 있다. 기타 netstat 이용 방법에 대해서는 man 페이지를 참고하기 바란다.

또는 tcpdump 를 사용할 수도 있다. Tcpdump 를 사용하여 특정 포트나 IP 에서의 패킷을 모니터링하여 패킷이 오가는지 여부를 점검하여 네트워크 장애시 문제 해결을 위해 활용할 수 있다. 이를테면 소스 IP 가 192.168.1.30 인 고객이 서버에 접속이 안 된다고 한다면 tcpdump src 192.168.1.30 으로 띄워 놓은 후 패킷이 오가는지 접속 여부를 모니터링해 볼 수 있을 것이다.

기타 tcpdump 는 아래와 같이 활용이 가능하다.

tcpdump port 80	# 80 번 포트로 오가는 패킷을 모니터링
tcpdump src 192.168.1.1	# 192.168.1.1 을 소스로 하는 패킷을 모니터링
tcpdump dst 211.3.4.5	# 211.3.4.5 를 목적지로 하는 패킷을 모니터링
tcpdump host 211.3.4.5	# 소스나 목적지가 211.3.4.5 인 패킷을 모니터링
tcpdump net 211.3.4.0/24	# 소스나 목적지가 211.3.4.0 대역 (즉 211.3.4.1 부터 211.3.4.254 까지)인 패킷을 모니터링

또한 tcpdump -a port 23 와 같이 사용할 경우에는 아래와 같이 telnet 접속을 스니핑 하는 용도로 사용될 수도 있다.

22:23:33.957216 eth0 > target.com.telnet > source.com.47034: P 83:90(7) ack 74 win 5792 <nop,nop,timestamp 34471591 12102054> (DF) [tos 0x10]

E^P ^@ ; .. w @^@ @^F r : .. / A j
.. / B 2 ^@^W .... a) ....
..^X ^V.. J ~ ^@^@ ^A^A ^H^J ^B^M ....
^@.. **.... l o g i n :**

22:24:15.101414 eth0 > target.com.telnet > source.com.47034: P 102:112(10) ack 84 win 5792 <nop,nop,timestamp 34475706 12106168> (DF) [tos 0x10]

E^P ^@ > .... @^@ @^F r - .. / A j
.. / B 2 ^@^W .... a < ....
..^X ^V.. .... ^@^@ ^A^A ^H^J ^B^N ^N..
^@.. .... **P a ss w o r d :**

tcpdump 이용에 대한 추가적인 정보에 대해서는 man 페이지나 http://www.tcpdump.org/ 를 참고하기 바란다. 추가적으로, 많이 사용하는 프로그램으로 ps 라는 명령어가 있다. Ps 는 Process Status 의 약자로 시스템의 프로세스 상태를 모니터링할 수 있는데 주로 ps auxw 로 옵션을 주어 사용하면 각종 정보를 모니터링 할 수 있다. 특히 어떤 프로세스가 cpu 나 메모리를 많이 사용하는지 등의 정보를 확인할 수 있다.

netstat 과 tcpdump 그리고 ps 의 명령어 사용법만 알고 있어도 대부분의 모니터링과 문제 발생시 해결을 할 수 있으며 이들 프로그램은 다른 툴의 기본이 되는 프로그램이므로 이 명령어의 사용법에 대해 확실히 숙지하는 것이 좋다.

포트 스캔 감지 프로그램 활용

원격지 시스템의 정보를 얻기 위해 공격자들이 제일 먼저 사용하는 방법이 바로 포트 스캔이다. 포트 스캔을 통해서 시스템의 열린 포트를 알아내어 어떤 서비스를 하고 있는지 알 수 있을 뿐만 아니라 운영체제, uptime 등의 정보도 얻을 수 있으며 이 정보를 토대로 시스템의 대략적인 보안 수준도 추측할 수 있다. 이 때문에 서버 관리자 입장에서 포트 스캔을 당했다 함은 차후에 침해를 받을 수 있는 가능성이 있다는 것을 뜻하므로 포트 스캔을 감지하고 이에 대해 대처할 수 있는 프로그램을 필요로 한다.

포트 스캔을 감지하거나 차단하는 프로그램은 많이 있는데, 그 중 RTSD 라는 프로그램을 추천한다. RTSD 는 Real Time Scan Detector 의 약자로 한국정보보호진흥원(certcc.or.kr)에서 공개한 프로그램인데, 실시간으로 클라이언트와 서버의 포트가 상호 반응하는 관계를 보여주거나 포트 스캔으로 감지시 해당하는 내용을 관리자에게 통보해 주기도 한다.

이 프로그램은 http://www.certcc.or.kr/cvirc/y2kvirus/down/RTSD/register.html 에서 다운로드 가능

하며 압축 해제후 Makefile 과 rule.h 파일을 수정해 주면 된다.
Makefile 에서는 SunOS 관련 설정인 LIBS = -lsocket -lnsl –lpcap 를 주석 처리하고 대신 주석되어 있는 리눅스의 #LIBS = -lpcap 주석을 삭제한다. 그리고 rule.h 파일에서는
#define IGNET "and not src net 192.168.1" 과 같이 스캔을 무시할 네트워크 대역을 정의하고
#define IGPORT "and not (port 80 or 110 or 20 or 21 or 25)" 와 같이 일반적으로 접속이 많아서 포트 스캔으로 감지하지 않을 포트를 or 로 나열해 주면 된다.
변경이 끝난후 make 를 실행하면 아래와 같이 몇 가지를 묻게 된다.

[기관명]:-> 에는 자신의 기관이나 상호명을 입력하고(예:오늘과내일)

[스캔공격 탐지를 보고받을 E-mail 주소(Default root)]:-> 는 포트 스캔 감지시 공격 탐지 정보를 받을 E-mail 주소를 기입하면 된다.(예:antihong@tt.co.kr)

[스캔공격 탐지를 CERTCC-KR 에 보고하시겠습니까(Y/N, Default Y)]:-> 에 Y 로 설정시에는 스캔 공격 정보를 CERTCC-KR 에도 함께 발송하여 포트스캔 공격 사실을 신고하게 된다.
이후 make 로 생성된 RTSD 실행 파일에 –d 옵션을 주어 ./RTSD –d 로 실행하면 백그라운드로 작동하게 된다.

아래는 실제 포트 스캔을 감지시 관리자에게 발송되는 메일 예이다.

[01/09/26 01:02:09]
[Scan Attack] From 192.168.1.2:34814 To 210.40.65.5:2503
[Scan Attack] From 192.168.1.2:34815 To 210.40.65.5:12294
[Scan Attack] From 192.168.1.2:34816 To 210.40.65.5:6813
[Scan Attack] From 192.168.1.2:34817 To 210.40.65.5:10197
[Scan Attack] From 192.168.1.2:34818 To 210.40.65.5:8144

RTSD 는 간단하면서도 훌륭한 기능을 제공하지만, 다른 포트 스캔 감지 프로그램처럼 Passive 모드에서의 FTP 데이터 전송을 포트 스캔으로 인식하는 문제가 있으며 엄격하게 설정을 적용하였을 경우 정상적인 접속도 포트 스캔으로 간주하는 경우가 있기도 하므로 각자의 상황에 따라 모니터링후 룰을 적당히 적용하여 사용하면 된다.
이외 Inetd 모드에서 작동하는 Klaxon (http://www.eng.auburn.edu/users/doug/second.html) 이나 RTSD 와 비슷한 기능을 가진 Scanlogd(http://www.openwall.com/scanlogd/) 그리고, 실시간으로 포트 스캔을 감지하여 Tcp Wrapper 나 ipchains 등을 이용하여 포트 스캔을 하는 소스 IP 를 차단하는 강력한 기능을 가진 PortSentry (http://www.psionic.com/abacus/portsentry/) 도 있다. 특히 PortSentry 의 경우 스캔을 하는 소스 IP 를 실시간으로 차단하는 기능이 있는데, 소스 IP 를 위조하여 스캔할 경우 문제가 될 수 있으니 정책 설정시 주의하여야 한다.

로그 모니터링 프로그램 활용

리눅스등 *NIX 계열이 Windows 계열에 비해 다른점은 로그에 대한 정책을 설정할 수 있고 많은 로그를 남길 수 있다는 점이다. 따라서 이 로그 정보를 효율적으로 모니터링하고 관리하는 것이 서버 관리자의 중요한 임무중 하나이다. 그러나 끝도 없이 쌓여가는 로그를 일일이 분석한다는 것은 거의 불가능에 가까우므로 이를 효과적으로 관리하여야 할 필요성이 제기된다. 이를 위해 로그를 관리해 주는 몇몇 프로그램에 대해 간단히 소개를 하겠다.
로그 모니터링 프로그램을 활용하기에 앞서 시스템에서 로그를 남기는 방식에 대해 먼저 이해를 하여야 하는데, 이에 대해서는 본지 8 월호의 Focus Part2 " 로그파일관리" 에 자세히 소개되었으므로 로그 관련 글을 먼저 참고하기 바란다.

Logcheck

Logcheck 는 실시간으로 로그 파일을 분석하여 사전에 해킹 시도등 비정상적인 상황이라고 정의된 내용이 로그에 남을 경우 해당 로그의 내용을 관리자에게 메일로 발송해 준다. 실시간 로그 체크는 logtail 이라는 프로그램을 이용하는데, 이 프로그램은 프로세스에 상주하여 실행되면서 체크한 로그 파일의 끝부분을 기억하고 있다가 새로운 로그의 내용이 추가될 때마다 계속적으로 로그 체크를 하게 된다.
Logcheck 는 http://www.psionic.com/abacus/logcheck 에서 다운로드 하여
압축 해제 후 압축이 풀린 디렉토리에서 make linux 를 실행하면 설정 파일과 실행 파일이 자동으로 /usr/local/etc/ 와 /usr/local/bin/ 디렉토리에 설치가 완료된다.
이후 /etc/crontab 파일에
00 * * * * root /bin/sh /usr/local/etc/logcheck.sh
와 같이 추가 설정하여 매 시간마다 logcheck 를 자동으로 실행하도록 한다.
이때 logcheck.sh 파일에서 SYSADMIN=antihong@tt.co.kr 부분을 이상 발생시 수신할
자신의 e-mail 주소로 변경해 주면 된다.
그리고

```
/etc/logcheck/logcheck.hacking          // 해킹공격 관련 메시지
/etc/logcheck/logcheck.ignore           // 해킹관련 로그에서 무시할 패턴
/etc/logcheck/logcheck.violations       // 부적절한 보안관련 메시지
/etc/logcheck/logcheck.violations.ignore // 부적절한 보안관련 로그에서 무시할 패턴
```

위 파일을 자신의 상황에 맞게 적절히 수정하여 사용하면 된다.

Swatch

Swatch 역시 Logcheck 와 비슷하게 로그 파일을 모니터링하다가 사용자가 지정한 패턴이 확인되었을 때 해당 내용을 메일로 발송하거나 벨소리를 내거나, 특정 스크립트를 실행하도록

할 수 있는 기능이 있으며 사용 방법에 대한 자세한 내용은 http://oit.ucsb.edu/~eta/swatch/ 를 참고하기 바란다.

Colorlog

이름에서도 알 수 있듯이 복잡한 로그 파일을 메시지의 내용에 따라 색을 다르게 하여 보여주는 프로그램으로서 사용 방법이 매우 쉽다.

http://www.resentment.org/projects/colorlogs/ 에서 관련 파일을 다운로드하여 압축해제 후 colorlogs.conf 에서 모니터링할 이벤트(메시지)와 색을 적절히 설정 후(기본설정을 그대로 사용해도 무방하다.) /etc/ 디렉토리로 옮긴 후

cat /var/log/messages | /usr/local/Colorlogs/colorlogs.pl 와 같이 실행하면 로그 파일에서 보안적으로 특별히 문제가 되는 부분만 다른 색으로 보여주어 로그 파일을 한눈에 쉽게 관리가 가능하다.

그리고 로그를 모니터링하여야 할 서버가 많을 경우 로그 서버를 별도로 운영하여 모든 로그를 한 서버에 몰아서 한꺼번에 관리할 수도 있다.

로그 서버를 운영하려면 먼저 로그를 보낼 서버의 /etc/syslog.conf 파일을 열어

authpriv.* @logserver.tt.co.kr 과 같이 설정후 syslogd 를 재실행하고

로그를 받을 로그 서버에서는 별도의 설정 변경 없이 기존의 syslogd 를 kill 한 후 **/usr/sbin/syslogd -m 0 -r - h** 를 실행해 주면 된다. 이후 로그 서버의 /var/log/secure 를 보면 아래와 같이 여러 서버의 로그가 한꺼번에 저장되는 것을 확인할 수 있다.

```
Sep  8 18:34:50 192.168.1.30 ipop3d[544]: connect from 211.47.64.90
Sep  8 18:34:50 192.168.1.72 ipop3d[29281]: connect from 211.107.196.139
Sep  8 18:34:50 192.168.3.2 ipop3d[30311]: connect from 211.118.155.221
Sep  8 18:34:50 192.168.3.55 ipop3d[25460]: connect from 211.47.64.90
Sep  8 18:34:51 192.168.2.100 ipop3d[27111]: connect from 128.134.0.8
```

위와 같이 원격지 서버의 IP 대신 호스트 이름이 남도록 하려면 로그 서버의 /etc/hosts 파일에

```
192.168.3.55          www50
192.168.3.2           www16
```

와 같이 호스트 이름을 정의해 주면 IP 대신 쉽게 구분이 가능한 호스트 이름으로 남게 된다.

```
Sep  8 18:34:50 www34 ipop3d[544]: connect from 211.47.64.90
Sep  8 18:34:50 www28 ipop3d[29281]: connect from 211.107.196.139
Sep  8 18:34:50 www16 ipop3d[30311]: connect from 211.118.155.221
Sep  8 18:34:50 www50 ipop3d[25460]: connect from 211.47.64.90
```

Sep 8 18:34:51 www26 ipop3d[27111]: connect from 128.134.0.8

백도어 점검하기

공격자들이 시스템을 공격하여 일정 권한을 획득한 후에는 차후에 다시 쉽게 들어올 수 있도록 자신만이 알고 있는 백도어(BackDoor) 를 만들어 두는 습성이 있다. 백도어는 여러 방식으로 구현 가능한데, 간단히 쉘을 복사해 두는 전통적인 방법부터 최근에는 커널 기반의 루트킷(RootKit) 까지 그 방식이 갈수록 교묘하고 고도화해지고 있어 백도어가 설치되어 있는지 여부를 점검하는 것이 매우 어렵다. 특히 보안에 그리 익숙하지 않은 초보 서버 관리자에게는 더더욱 그러한데, 이를 위해 아주 쉬우면서도 유용한 백도어 점검 프로그램인 chkrootkit 이라는 프로그램을 소개하고자 한다. 이 프로그램은 호스트내에서 t0rn 이나 라면 바이러스, 라이온 바이러스등 20 여가지의 최근의 각종 루트킷 설치 여부를 점검해 주며 ifconfig, ps , netstat 등 변조되기 쉬운 40 여가지의 바이너리 프로그램의 변조 여부를 체크해 준다.

이 프로그램을 이용하려면 먼저 http://www.chkrootkit.org/ 에서 최신 버전의 tarball 파일을 다운로드하여 압축 해제후 압축 해제한 디렉토리에서 make sense 를 실행하여 컴파일해 주기만 하면 모든 설치는 완료된다.

체크 방법은 두 가지로 할 수 있다. 첫번째는 chkrootkit 이 설치된 디렉토리에서

./chkrootkit 만 실행해 주면 자동으로 시스템을 검색하여 루트킷이나 파일 변조 여부를 알려준다. 좀 더 세밀한 분석을 하고자 할 경우에는 # ./chkrootkit -x | more 와 같이 strings 를 이용하여 바이너리 파일을 직접 분석할 수도 있다.

또한 ./chkproc -v 로 Hidden Process 여부도 확인 가능하다. Hidden Process 란 실제 프로세스에 떠서 작동하고는 있지만 ps auxw 등으로는 보이지 않는 것으로 이는 ps 에서 보이는 프로세스 정보와 /proc/pid/ 정보와 비교하여 ps 에서 보이지 않는 숨겨진 프로세스가 있는지 여부를 조사하는 방법으로 매우 유용하다. 하지만 짧은 시간에 많은 접속이 있는 웹서버등의 시스템의 경우 짧은 시간에 많은 프로세스가 죽었다 살았다를 반복하므로 잘못된 정보가 출력될 수도 있으므로 여러 번 실행해 보아야 한다.

파일 시스템 무결성 감시 프로그램

Fcheck

현재의 파일 시스템이 무결한 상태라고 판단하였을 경우 파일이나 파일 시스템의 무결성을 체크하기 위해 매번 위 프로그램을 실행할 수는 없다. 또한 chkrootkit 의 경우 일부 프로그램에 대해서만 체크를 하므로 상시적으로 모든 파일 시스템의 무결성을 체크하는 모니터링 프로그램을 필요로 하게 된다. 이러한 목적으로 파일이나 파일 시스템의 무결성을 모니터링 할 수 있는 프로그램으로는 Tripwire 가 가장 많이 알려져 있다. Tripwire 설정예는 본지 9 월호에서 다루어졌으므로 여기에서는 Tripwire 에 비해 간단한 설정으로도 막강한 기능을

제공하는 Fcheck 라는 프로그램을 소개하고자 한다.
Fcheck 는 지정한 디렉토리나 파일에 대한 각종 정보를 데이터베이스로 보관하고 있다가 정해진 시간마다 시스템을 체크하여 현재의 데이터 베이스 정보와 비교하여 파일 변조 여부를 모니터링하는 프로그램으로 빠른 속도와 간단한 사용 방법으로 사용을 권장하는 프로그램이다. Fcheck 는 http://www.geocities.com/fcheck2000/fcheck.html 에서 다운로드하여 압축 해제후 fcheck 라는 실행 파일과 fcheck..cfg 라는 설정파일 이렇게 두개 파일만 수정해 주면 된다. 먼저 fcheck 파일에서 $config 부분을 **$config="/usr/local/fcheck/fcheck.cfg";** 와 같이 fcheck 가 설치된 디렉토리로 변경하고

fcheck.cfg 파일을 열어

```
Directory = /bin/          #  무결성을 모니터링할 디렉토리
Directory = /usr/bin/      #  무결성을 모니터링할 디렉토리
Exclusion                  #  모니터링을 제외할 디렉토리나 파일
Database =  /usr/local/fcheck/data/data.dbf     # 파일 시스템에 대한
                                                  데이터 베이스가 저장될 파일의 경로
TimeZone= GMT-9        # Timezone 설정
```

과 같이 설정후 fcheck - ac 를 한번 실행해 주면 현재의 파일 시스템 정보를 데이터베이스화 하여 지정한 파일에 저장한다. 현재의 파일 시스템 데이터베이스 정보를 근거로 이후에 변경되는 파일 시스템의 정보를 비교하므로 현재의 파일 시스템은 무결해야 한다.
아울러 /var 등과 같이 로그 파일이 있어 파일 시스템 정보가 자주 변경되는 파티션이나 파일은 모니터링할 디렉토리에 지정하지 않는 것이 좋다. 이후 fcheck - a 를 실행하면 현재의 파일 시스템 정보와 비교하여 추가나 삭제 또는 변경된 내용이 있을 경우 변경된 내용을 알려주고, 패키지 업데이트등 변경된 정보가 정상적인 경우에는 fcheck - ac 로 데이터 베이스를 업데이트 하여야 한다.
아래는 /usr/bin 에 .hack 이라는 파일을 생성 후 fcheck - a 를 실행해 본 결과이다.

```
ADDITION: [www.tt.co.kr] /usr/bin/.hack
        Inode    Permissons        Size      Created On
        275505   -rw-r--r--        0         Sep 06 01:28 2001
```

위와 같이 파일이 추가되었음을 알 수 있다. 파일의 정보가 변경시에는 ADDITION 대신 WARNING 이라는 메시지가 뜨고 파일이 삭제되었을 경우에는 DELETION 이라고 나타난다.
fcheck 에는 자동 메일통보등의 기능이 없으므로 이 작업을 매일 자동으로 하고 싶으면 /etc/cron.daily/ 디렉토리에 아래와 같은 실행 스크립트를 뱔도로 설정하여 실행하면 된다.

```
#!/usr/bin/perl

$TASK = `/usr/local/fcheck/fcheck -a|grep Inode`;  # 설치한 자신의 경로에 맞게 설정.
$HOSTNAME = `/bin/hostname`;
$TO_MAIL        = 'antihong@tt.co.kr';    # 이상 발견시 통보를 받을 e-mail 주소를 입력.
$SUBJECT        = "$HOSTNAME File 변조 확인";  # 통보받을 메일의 제목을 지정.
$MAIL_PROGRAM   = "/usr/sbin/sendmail";
if ($TASK){
  $TASK_CONFIRM = `/usr/local/fcheck/fcheck -a`;  #  자신의 경로에 맞게 설정.
&task_confirm;
}
sub task_confirm{
open(MAIL, "|$MAIL_PROGRAM -t");
      print MAIL "To: $TO_MAIL \n";
      print MAIL "Subject: $SUBJECT \n\n";
      print MAIL "Host: $HOSTNAME \n";
      print MAIL "$TASK_CONFIRM \n";
close(MAIL);
 }
```

위와 같이 설정 후에는 매일 자동으로 파일 시스템의 무결성을 체크하여 변경된 정보가 있을 경우에는 변경된 내용을 관리자에게 메일로 통보해 주게 된다.
만약 통보된 변화가 정상적인 것이라면 “fcheck –ac” 명령을 이용하여 데이터 베이스를 새로운 정보로 업데이트 하면 된다.

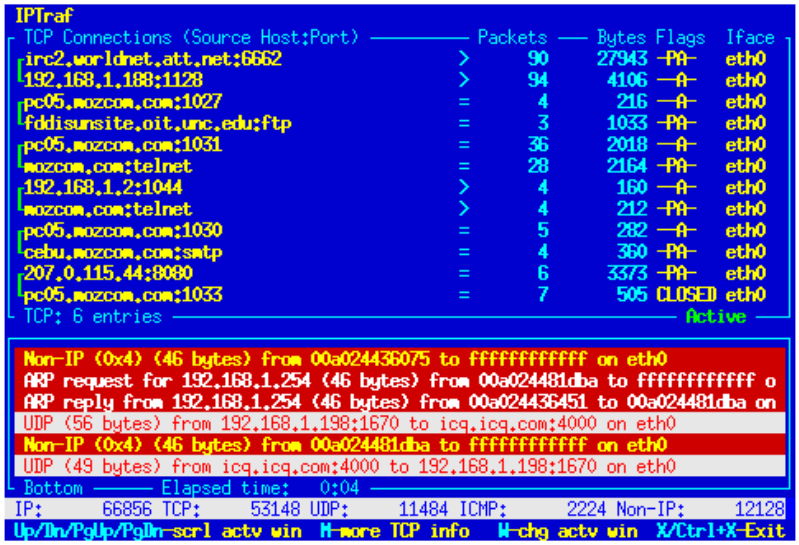
sXid

suid 나 sgid 등 s 비트가 설정된 디렉토리나 파일은 보안상 매우 중요하다는 것을 잘 알고 있을 것이다. 따라서 suid 나 sgid 가 설정된 파일이나 디렉토리의 무결성에 대해서만 별도로 모니터링하는 프로그램을 필요로 한다면 sXid 라는 프로그램을 추천한다.
이 프로그램은 위에서 설명한 fcheck 와 작동하는 방식이 유사하다. cron 을 이용하여 정기적으로 suid/sgid 를 모니터링하여 기본적으로 시스템에서 suid 나 sgid 가 설정되어 있는 디렉토리나 파일에 어떤 변화가 있는지 체크를 한다. 파일이나 디렉토리가 새롭게 생기거나 또는 비트나 다른 모드를 변경했을 경우 sXid 는 그 변화에 대해 e-mail 등으로 보고를 하게 된다. 일단 설치가 되면 자동으로 모든 일을 처리하므로 일일이 신경 쓰지 않아도 된다.

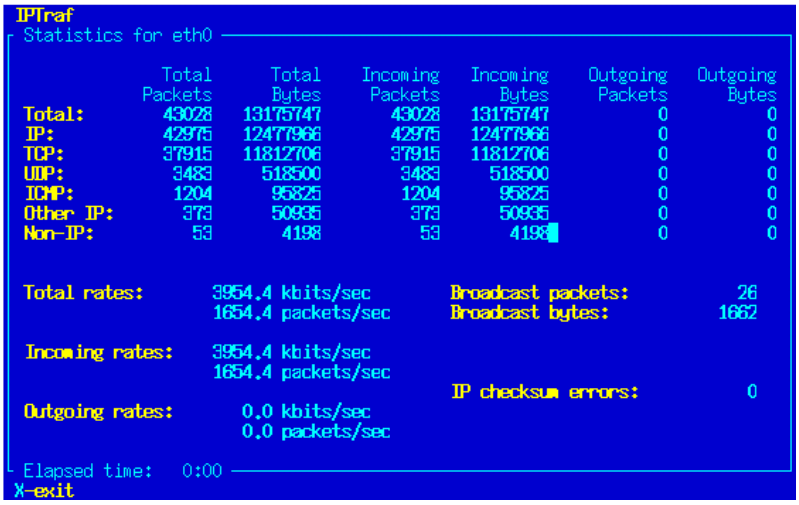
먼저 ftp://marcus.seva.net/pub/sxid/sxid_4.0.1.tar.gz 에서 파일을 다운로드 받아
압축 해제후 압축해제된 디렉토리에서 ./configure ; make ; make install 를 실행하면 설치가 완료된다.
실행 파일은 /usr/local/bin/sxid 이고 설정 파일은 /usr/local/etc/sxid.conf 인데, 특별히 변경할 부분은 없고 단지 EMAIL = " antihong@tt.co.kr" 부분만 정보 변경시 통지를 받을 자신의 e-mail 로 변경하면 된다.
그리고 /etc/cron.daily/ 디렉토리에
#/bin/sh
/usr/local/bin/sxid
와 같이 파일을 만들어 두면 매일 자동으로 sxid 를 실행하여 모니터링 결과를 메일로 발송해 준다.

패킷 모니터링 프로그램

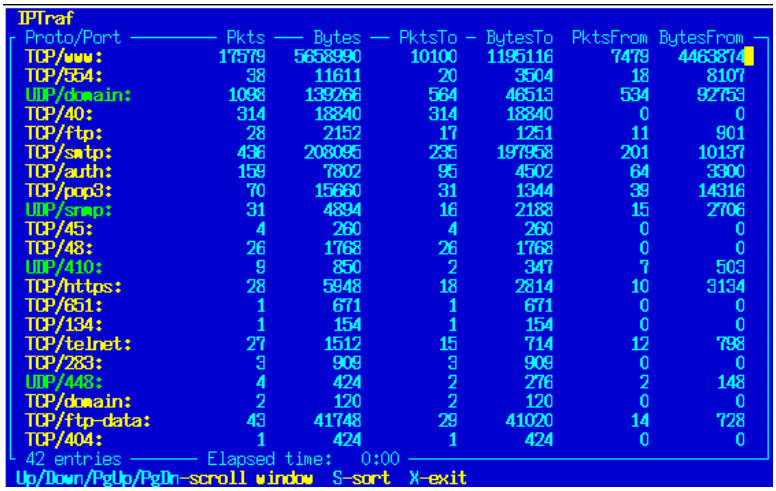
앞에서 tcpdump 나 netstat 등을 이용하여 간단히 패킷이나 세션을 모니터링하는 방법에 대해 알아보았다. 그러나 위 프로그램의 경우 직관적으로 패킷 모니터링을 하기에는 한계가 많으므로 iptraf 라는 별도의 패킷 모니터링 프로그램을 추천하고자 한다.
Iptraf 는 작고 가벼우면서도 강력한 기능을 제공하는 프로그램으로서 네트워크에서 나가고 들어오는 모든 요청을 실시간으로 모니터링하는데, 각 호스트, 포트, 프로토콜등에 대한 세부적인 정보를 제공한다. Iptraf 는 http://iptraf.seul.org/ 에서 파일을 다운로드후 압축을 해제하여 ./Setup 만 실행해 주면 설치가 완료된다.
설치된 디렉토리에서 iptraf 를 바로 실행하면 IP 트래픽, 인터페이스 통계, LAN station 모니터등 메뉴식으로 각종 정보를 쉽게 볼 수 있으며 프로토콜별로 필터링 기능도 제공한다.
iptraf -d eth0 -B 와 같이 Background 로 작동시에는 기본적으로 /var/log/iptraf 디렉토리에 매 5 분마다 각 프로토콜의 사용량, 트래픽등의 정보가 출력된다.



< 그림 3. iptraf IP Traffic 모니터 >



< 그림 4. iptraf 를 이용한 프로토콜별 통계>



< 그림 5. iptraf 를 이용한 Port 별 통계>

가상 호스팅 도메인 모니터링

한 서버에 여러 개의 도메인을 설치하여 웹호스팅을 하는 경우 각각의 도메인에 대해 트래픽이나 하루 접속자 수등을 모니터링 해야 할 필요성이 제기된다. 특히 웹호스팅을 전문적으로 하는 업체에서는 필수 기능이라 할 수 있는데도 불구하고 실제로 이 방법이 잘 알려져 있지 않아 일일이 수작업으로 모니터링을 하거나 아예 모니터링을 포기해야 하는 경우가 많다. 한 서버 전체에 대해 트래픽을 관리하려면 서버에 snmpd 데몬을 띄우고 mrtg 를 설치하여 모니터링하면 된다. 이 방법은 본지에서도 여러 번 소개되었으므로 관련 기사(본지 8 월호 focus) 를 참고하시기 바란다. 아울러 한 서버에 여러 도메인을 설치하여 가상 웹호스팅을 서비스 하고 있는 경우 각각의 도메인에 대해 트래픽을 관리할 수 있는 방법은 몇 가지가 있지만 일일이 직접 프로그램을 코딩해야 하므로 가장 쉽고 효과적인 방법으로 mod_throttle 이라는 아파치 모듈을 이용하는 방법을 추천한다. 이 아파치 모듈은 가상 호스트별로, 디렉토리별로, 위치(Locations)별로 또는 유저에 따라 서버의 로드를 낮추거나 대역폭을 낮추기 위해 개발되었다. 이를 통해 각각의 가상 호스팅 도메인에 대해 접속 속도를 조절하거나 일정 제한을 초과할 경우에는 접속 요청을 거부할 수도 있다. 설치를 하려면 아파치 웹서버에 모듈로 포함시키거나 정적으로 포함시켜야 하는데, 결국 아파치 컴파일을 다시 하여야 한다.

먼저 http://www.snert.com/Software/mod_throttle/mod_throttle312.tgz 에서 관련 소스 파일을 다운로드하여 압축을 해제한다.

DSO 모듈을 빌드하기 위해서는 아파치 소스 디렉토리에서 아래와 같이 한다.

```
cd (path to)/mod_throttle-3.1
make install
```

권장하는 방법으로, 스태틱하게 모듈을 빌드하려면 아파치 소스 디렉토리에서 컴파일시 아래와 같이 mod_throttle 를 모듈에 추가하도록 한다.

```
./configure --prefix=/usr/local/apache --activate-module=src/modules/php4/libphp4.a
--add-module=../mod_throttle-3.1.2/mod_throttle.c
```

위 설정은 php4 와 mod_throttle 를 아파치에 모듈로 설정하는 화면이다.

아파치 컴파일을 완료한 후 생성된 /usr/local/apache/bin 디렉토리에서 httpd -l 을 실행하여 mod_throttle.c 이 모듈에 포함되었는지 여부를 확인한다. 모듈로 포함되었음이 확인되면 일단 설치는 완료되었으니 이제 본격적으로 httpd.conf 에 설정을 해 보도록 하겠다.

```
# 공통설정부분
<Location /throttle-status>
   Order deny,allow
   Deny from all
   Allow from 192.168.1.0/255.255.255.0
</Location>
```

위는 throttle-status 를 설정하였을 경우 http://domain.com/throttle-status/ 로 접속시 모든 도메인에 대한 트래픽이나 접속자수등의 중요한 정보가 그대로 노출되므로 throttlee-status 에 대해 특정 IP 대역에서만 (위의 경우 192.168.1.0 대역에서만) 접근이 가능하도록 제한하는 설정이다.

```
# 일반 설정 부분.
<IfModule mod_throttle.c>
      ThrottlePolicy None              // 기본적으로는 None 으로 설정한다.
   <Location /throttle-status>
      SetHandler throttle-status       // throttle-me 등도 있는데, throttle-status 만 필요하다.
   </Location>
</IfModule>
```

참고로 throttle-me 는 http://domain.com/throttle-me/ 로 접속시 domain.com 이라는 하나의 도메인에 대해서만 정보를 확인할 수 있도록 하는 방법이다. http://domain.com/throttle-status/ 로 접속시에는 모든 가상 도메인에 대해 정보를 볼 수 있게 된다.

```
# 가상 호스트 설정 부분.
<VirtualHost data1.tt.co.kr)
   ServerAdmin antihong@tt.co.kr
   DocumentRoot /home/data/public_html
   ServerName data1.tt.co.kr
   ThrottlePolicy Volume 300M 1d          //하루에 전송량 300M
   ThrottlePolicy Request 1000 1d         //하루에 히트수 1000 회 제한
   ThrottleClientIP 100 volume 200 300    // 로그를 100K 남기며 300 초간 200K 의 전송량 제한
</VirtualHost>
```

위와 같이 설정 후 전송량이나 히트수등을 초과하면 data1.tt.co.kr 접속시 원래의 페이지가 아닌 503 에러 화면이 뜨게 되며, ErrorDocument 503 /~user/ 로 redirect 설정을 추가해 주면 트래픽 초과시 503 에러화면 대신 /~user/index.html 파일을 보여 주도록 한다. 따라서 이 파

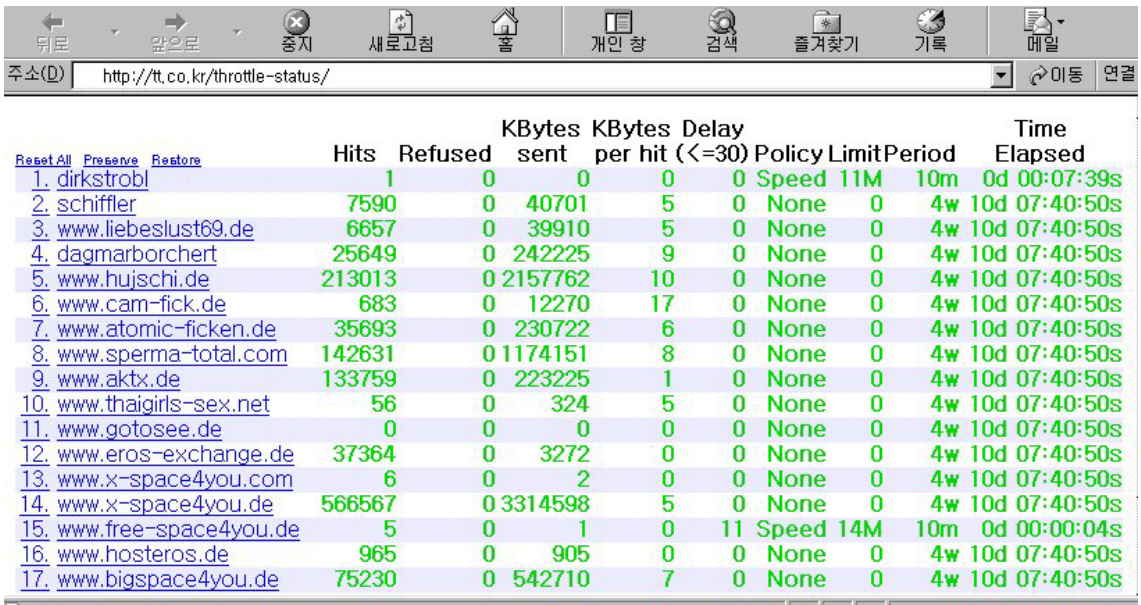
일에 트래픽이나 히트수등의 초과에 대한 경고문등 적당한 메시지 화면을 만들면 된다.

그리고 아래와 같이 설정시에는,

```
# 일반 설정 부분.
<IfModule mod_throttle.c>
      ThrottlePolicy Volume 300M 1d
      <Location /throttle-status>
      SetHandler throttle-status
      </Location>
</IfModule>

  # 가상 호스트 설정 부분.
<VirtualHost data2.tt.co.kr)
   ServerAdmin antihong@tt.co.kr
   DocumentRoot /home/data2/public_html
   ServerName data2.tt.co.kr
   ThrottlePolicy Volume 500M 1d       // 하루에 전송량 500M
</VirtualHost>
```

위와 같이 설정시 별도로 제한을 설정하지 않은 모든 도메인은 하루 전송량이 1 일 300 메가로 제한되지만 data2.tt.co.kr 은 500 메가로 제한된다.
모든 설정에 대한 관리는 http://www.mydomain.com/throttle-status 에서 가능하며
보다 자세한 이용 방법은 http://www.snert.com/Software/mod_throttle/ 를 참고하기 바란다.
또한 http://www.snert.com/Software/mod_watch/ 모듈을 이용하면 이 기능외에 각 가상 도메인 별로 인바운드, 아웃 바운드되는 트래픽을 mrtg 그래프로도 볼 수 있다.



Reset All Preserve Restore	Hits	Refused	KBytes sent	KBytes per hit	Delay (<=30)	Policy	Limit	Period	Time Elapsed
1. dirkstrobl	1	0	0	0	0	Speed	11M	10m	0d 00:07:39s
2. schiffler	7590	0	40701	5	0	None	0	4w	10d 07:40:50s
3. www.liebeslust69.de	6657	0	39910	5	0	None	0	4w	10d 07:40:50s
4. dagmarborchert	25649	0	242225	9	0	None	0	4w	10d 07:40:50s
5. www.hujschi.de	213013	0	2157762	10	0	None	0	4w	10d 07:40:50s
6. www.cam-fick.de	683	0	12270	17	0	None	0	4w	10d 07:40:50s
7. www.atomic-ficken.de	35693	0	230722	6	0	None	0	4w	10d 07:40:50s
8. www.sperma-total.com	142631	0	1174151	8	0	None	0	4w	10d 07:40:50s
9. www.aktx.de	133759	0	223225	1	0	None	0	4w	10d 07:40:50s
10. www.thaigirls-sex.net	56	0	324	5	0	None	0	4w	10d 07:40:50s
11. www.gotosee.de	0	0	0	0	0	None	0	4w	10d 07:40:50s
12. www.eros-exchange.de	37364	0	3272	0	0	None	0	4w	10d 07:40:50s
13. www.x-space4you.com	6	0	2	0	0	None	0	4w	10d 07:40:50s
14. www.x-space4you.de	566567	0	3314598	5	0	None	0	4w	10d 07:40:50s
15. www.free-space4you.de	5	0	1	0	11	Speed	14M	10m	0d 00:00:04s
16. www.hosteros.de	965	0	905	0	0	None	0	4w	10d 07:40:50s
17. www.bigspace4you.de	75230	0	542710	7	0	None	0	4w	10d 07:40:50s

< 그림 5. 실제 throttle-status 작동 화면 >

3.외부 시스템(네트워크) 모니터링 소프트웨어 활용

호스트 기반의 보안이 운영체제에 의존하여 하나의 호스트에 대한 각종 보안 정책 및 모니터링 설정을 의미하는 것이라면 네트워크 보안은 운영 체제에 관계없이 네트워크 패킷에 대한 정책 설정을 의미한다. 이 파트에서는 이 개념 뿐만이 아니라 좀 더 넓은 의미의 외부 시스템 보안을 포함하여 각종 네트워크 모니터링 방법에 대해 알아보도록 하자.

스니핑(Sniffing) 모니터링

동일 네트워크에 대형의 시스템들이 몰려있는 웹호스팅 업체나 IDC 등에서는 스니핑 공격에 매우 유의하여야 한다. 취약한 하나의 시스템에서만 관리자 권한을 획득하여 스니핑을 돌린다면 전체 네트워크를 장악하는 것은 시간 문제이기 때문이다.
일반적으로 스니핑이 작동하고 있다면 인터페이스 카드가 PROMISC 모드(우리말로 무차별 모드라고 한다.) 로 변하게 된다. Promisc 모드로 설정되어야 랜 상의 모든 트래픽을 받아들일 수 있기 때문이다. 시스템 내부에서 스니핑이 작동하고 있는지의 여부는 단순히 ifconfig -a|grep PROMISC 로 이더넷 카드에 PROMISC 가 설정되어 있는지를 쉽게 확인할 수 있으나 시스템 외부에서 그 여부를 확인하기는 쉽지 않다.
대부분의 문서에서는 인터페이스 카드가 Promisc 로 설정되어 있다면 스니핑이 돌고 있다고 밝히고 있지만, 실제로 시스템에 따라 기본적으로 Promisc 로 설정되는 경우도 있으므로 PROSMIC 로 설정되어 있다고 해서 반드시 스니핑이 작동하고 있는 것은 아니다.
네트워크에서 스니핑이 작동하는지 여부를 모니터링 할 수 있는 몇 가지 프로그램이 있는데, 추천하고 싶은 프로그램으로 Sentinel 이나 Antisniff 그리고 관련 프로그램으로 ARPWATCH 가 있다.
Sentinel 을 설치하려면 미리 패킷 캡처 라이브러리인 Libnet 1.0 과 libpcap 가 설치되어야 하는데, 각각 (http://www.packetfactory.net/Projects/libnet) 와 (ftp://ftp.ee.lbl.gov/libpcap-0.4.tar.Z) 에서 다운로드 받아 압축 해제 후 압축이 풀린 디렉토리에서 ./configure; make ; make install 로 설치하면 된다. 그리고 Sentinel 은 http://www.packetfactory.net/Projects/sentinel/ 에서 다운로드 가능하며 다운로드하여 압축 해제 후 압축이 풀린 디렉토리에서 make all 로 컴파일하

여 설치하면 된다. 현재 Sentinel 이 원격지 시스템에서 스니핑이 작동하는지 여부를 감지하는 방법은 3 가지가 있는데, 이 방법은 각각 DNS test, Etherping test, ARP test 이다.

먼저 각각의 방법이 가능한 원리에 대해 간단히 살펴보면, 우선 DNS test 의 경우 목적지 서버에 위조된 연결 요청을 보내어, 일반적인 스니핑 프로그램이 요청받은 시스템의 IP 주소를 역리졸브(Inverse DNS lookup) 한다는 특징을 이용하여 DNS 트래픽을 감시하여 스니핑을 감지하는 방법이다.
Etherping test 는 목적지에 ping 을 보낼 때 목적지의 IP 는 맞지만 목적지의 MAC 주소는 존재하지 않는 정보로 하여 Icmp Echo Packet 을 보내어 응답이 오는지를 감시하는 방법으로 대부분의 정상적인 시스템에서는 MAC 정보가 올바르지 않기 때문에 패킷을 DROP 하지만 promiscuous mode 가 설정된 시스템에서는 응답을 한다는 특징을 이용하여 감시하는 방법이다. 마찬가지로 ARP test 역시 IP 는 맞지만 목적지의 MAC 주소를 다르게 하여 ARP 요구를 보내는 방법으로 Promisc 모드가 아닌 경우에는 패킷이 목적지까지 갈 수 없으므로 목적지에서는 응답하지 않지만 Promisc 모드인 경우에는 모든 패킷을 받아들이므로 결국 응답한다는 특징을 이용하여 감시하는 방법이다.
각각의 방식에 대한 실행 방법예는 아래와 같다.

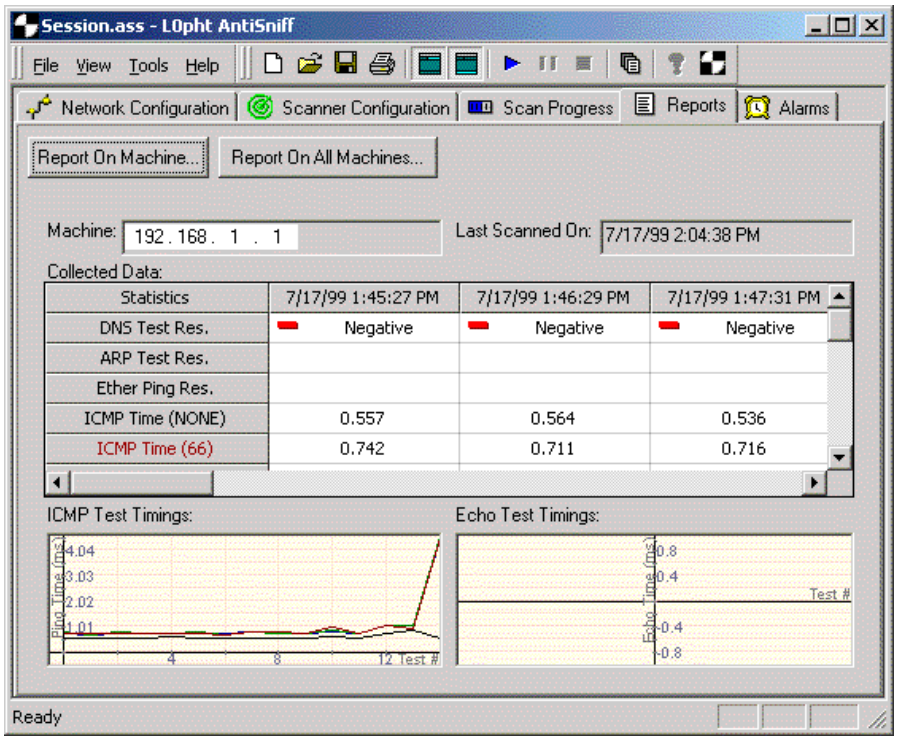
```
./sentinel -a -t 192.168.1.2                # ARP 테스트
./sentinel -d -f 1.1.1.1 -t 192.168.1.2     # DNS 테스트
./sentinel -e -t 192.168.1.2                # Etherping 테스트
./sentinel -t 192.168.1.2 -f 1.1.1.1 -d -a –e   # 3 개의 테스트를 동시에 수행
```

위와 같이 실행시
Results: 192.168.1.2 tested positive to etherping test.
와 같이 탐지 결과가 positive 가 나오면 Promisc 모드로 설정되었다는 의미이므로 해당 인터페이스의 PROMISC 여부를 조사하여야 한다.

아울러 Antisniff 의 경우 http://www.securitysoftwaretech.com/antisniff/ 에서 다운로드 가능하며 리눅스 버전과 아울러 윈도우 버전의 프로그램도 이용 가능하다. 테스트 방법은 위의 Sentinel 과 비슷하며 추가적으로 Latency test 라는 방법이 있는데, 이 방법은 스니핑이 작동시 무차별 모드로 설정되어 있어 로컬 네트워크상의 모든 트래픽을 받아들이느라 로드가 높아진다는 점을 이용해 불필요한 트래픽을 전송하여 시스템의 응답 시간이 길어지는지 여부를 조회하는 재미있는(?) 방법으로 아직까지는 100% 신뢰할 수는 없는 방법이며 계속적으로 개선중인 기능이다.

윈도우 버전의 경우 테스트하는 IP 대역을 지정하여 한꺼번에 검사가 가능하고 검사 결과

에 대해 각종 통계도 볼 수 있으며, 얼마의 주기로 테스트를 할 것인지를 시간대별, 날짜별, 주별로 정할 수 있는 예약 기능 및 검사 결과가 변경시 메일로 발송하거나 음악이 나오게 하는 등의 알람 기능도 있어 편리하게 사용이 가능하다.



< 그림 6. Antisniff 를 이용한 체크결과 화면>

끝으로 이와 관련하여 권장할 만한 프로그램으로 ARPWATCH 가 있다.

이 프로그램은 ARP 트래픽을 모니터링하여 MAC 주소와 IP 간 매칭을 감시하는 프로그램으로 정보가 변경되거나 새로운 MAC 주소가 확인시에는 해당하는 내용을 관리자에게 메일로 통보하게 된다. ARPWATCH 를 이용시 MAC 주소나 ARP 를 이용하는 공격에 대한 대응에 매우 유용하다. ARPWATCH 는 패킷 캡처 라이브러리인 libpcap 를 사용하므로 먼저 ftp://ftp.ee.lbl.gov/libpcap.tar.Z 파일을 다운로드하여 압축 해제후 ./configure; make ; make install 로 libcap 프로그램을 설치한다. 잘 설치되지 않으면 http://rpmfind.net/ 에서 rpm 으로 다운로드하여 설치도 가능하다. 라이브러리가 설치되면 ftp://ftp.ee.lbl.gov/ 에서 arpwatch 프로그램을 다운로드 하여 압축 해제 후 설치 전에 addresses.h 파일을 열어

#define WATCHER "antihong@tt.co.kr"

#define WATCHEE "arpwatch (Arpwatch)"

와 같이 변경된 ARP 정보 발견시 메일을 수신할 e-mail 주소와 발송시 발신자 정보를 정의

한다. 변경 후 압축 해제한 디렉토리에서 ../configure; make ; make install 로 설치를 하면 된다. 설치 이후 같은 디렉토리에서 touch arp.dat 로 파일을 생성 후 ./arpwatch –d 를 실행하면 현재 네트워크의 MAC 정보와 IP 정보를 매칭하여 arp.dat 파일에 데이터 베이스쌍을 생성한다. 데이터 베이스 생성이 완료된 후 ./arpwatch 를 실행하여 두면 실시간으로 ARP 트래픽을 체크하여 새로운 MAC 정보가 추가되거나 MAC 정보가 변경되는 등 현재의 데이터 베이스 정보와 다른 정보가 발견 시에는 앞에서 지정한 관리자에게 해당하는 내용에 대해 통보를 하고 데이터 베이스를 갱신하게 된다.
아래는 MAC 정보가 변경시 관리자에게 메일로 통보된 내용의 예이다.

hostname: arp.tt.co.kr
ip address: 192.168.1.1
ethernet address: 0:0:1c:2:38:18
ethernet vendor: JDR Microdevices generic, NE2000 drivers
old ethernet address: 0:50:bf:30:fe:50
old ethernet vendor: <unknown>
timestamp: Monday, April 23, 2001 21:49:08 +0900
previous timestamp: Monday, April 23, 2001 20:11:47 +0900
delta: 1 hour

네트워크에서 특정 문자열 탐지하기

얼마전까지 많은 서버에 피해를 주었던 코드레드 바이러스는 Windows NT 나 Windows 2000 의 IIS 서버만 공격하는 것으로 알려져 있지만 실제로 웹서버의 버전과는 관계 없이 80 번 포트로 무차별적인 접속시도를 하여 웹 기반의 스위칭이나 라우터등 일부 네트워크 장비가 다운 되는 등의 문제가 있었다. 또한 아파치 서버의 경우 로그를 남겨 놓았을 경우 서버에 많은 로그를 남기어 디스크가 Full 이 되는 경우도 있었다. 코드 레드의 경우 각 서버의 로그 파일을 보면 공격지 IP 를 확인할 수 있지만 일일이 각 서버의 로그파일을 남기거나 분석하지 않고도 네트워크상에서 특정 문자열로 탐지가 가능하다.
이는 ngrep 이라는 툴을 이용하면 된다. ngrep 은 네트워크에 전송되는 트래픽에서 특정 문자열이나 텍스트만을 검색하는 막강한 기능을 가진 프로그램으로 코드 레드의 경우 기본적으로 default.ida 파일을 요구하므로 이 문자열을 요구하는 패킷을 검색하면 된다.

```
# ngrep -qt ".ida\?" tcp port 80
2001/09/05 18:50:34.514746 211.192.184.151:3441 -> 211.40.15.176:80 [A]
GET /default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

위의 경우 211.192.184.151 에서 211.40.15.176 번 서버로 코드레드 시도를 하고 있는 것을 알 수 있다. 이처럼 ngrep 은 네트워크상의 패킷에서 특정 문자열만을 뽑아냄으로써 이를 응용한다면 모든 네트워크 모니터링 프로그램이 그러하듯이 막강한 스니핑 툴로도 악용이 가능하다. 만약 네트워크상에서

 # ngrep -wi 'user|pass' tcp port 110 와 같이 입력하였다면 잠시 후 아래와 같이

interface: eth0 (211.40.165.43/255.255.255.0)

filter: ip and (tcp port 110)

match: ((^user|pass\W)|(\Wuser|pass$)|(\Wuser|pass\W))

T 61.73.202.236:26129 -> 211.40.165.43:110 [AP]

USER admin..

T 61.73.202.236:26129 -> 211.40.165.43:110 [AP]

PASS qwert!23a..

pop3 로 접속하는 Id /pw 가 그대로 노출되는 것을 확인할 수 있을 것이다. 물론 위와 같이 110 번 외에 다른 Port 의 스니핑도 가능하다. Ngrep 은 패킷 라이브러리를 사용하므로 프로그램 설치에 앞서 앞에서 설명한 libpcap 을 먼저 설치한 후 http://www.packetfactory.net/projects/ngrep/ 에서 ngrep 을 다운로드하여 압축 해제후 압축 해제한 디렉토리에서 ./configure; make 로 설치하면 된다. 또는 http://rpmfind.net/ 에서 rpm 형태의 파일을 직접 다운로드하여 설치하여도 된다.

ngrep -q 'sex|porno' 와 같이 실행 시에는 네트워크내 패킷에서 sex 나 porno 라는 문자열이 포함된 트래픽을 찾아주는 등 여러 가지로 활용이 가능하니 구체적인 이용에 대한 안내는 홈페이지를 참고하기 바란다.

mrtg 로 메일서버 모니터링 하기

mrtg 를 이용하여 서버나 라우터의 트래픽이나 데이터 전송량을 모니터링 하는 방법은 많이 소개되었으니 이번에는 메일 서버에서 각종 정보를 mrtg 로 통계내어 확인하는 방법을 살펴보도록 하자. 일단 메일 서버의 각종 통계 데이터는 sendmail 에서 제공하는 mailstats 라는 프로그램을 이용하면 되는데, mailstats 는 /var/log/sendmail.st 파일을 참고하므로 sendmail.cf 에서 위 설정이 주석 처리되지는 않았는지 확인하여 주석이 되어있으면 주석을 제거하고, /var/log 디렉토리에 sendmail.st 파일을 생성한 후 sendmail 을 재시작해 주면 이 파일에 관련 정보가 생성된다.

그리고 메일서버의 정보를 mrtg 로 보려면 mrtg-mailstatus 라는 별도의 프로그램을 필요로 하는데, 이 프로그램은 http://www.infocom.com/~littell/software/mrtg-mailstats.html 에서 관련 파일을 다운로드하여 설치 후 아래와 같이 mrtg 를 위한 cfg 파일을 만든 후 mrtg 로 실행해 주면 된다.

```
# 이 설정은 하루에 전송되는 smtp.tt.co.kr 서버의 메일개수를 mrtg 로 보여준다.

Target[mail.tt.co.kr]:`/usr/local/mrtg/mrtg-mailstats -s smtp.tt.co.kr`
Title[mail.tt.co.kr]: E-mail 개수
YLegend[mail.tt.co.kr]: Sent mail number

# 이 설정은 smtp.tt.co.kr 서버의 하루전송 메일 트래픽을 mrtg 로 보여준다.
Target[mail.tt.co.kr]: `/usr/local/mrtg/mrtg-mailstats -b -s smtp.tt.co.kr -p 2525`
MaxBytes[mail.tt.co.kr]: 1250000
Title[mail.tt.co.kr]: Remote Mail Traffic
PageTop[mail.tt.co.kr]: <H1>Remote Mail Traffic</H1>
```

그리고 메일 서버에서는 /etc/services 파일에

smtp-stats 2525/tcp snmp-stats 를 추가하고

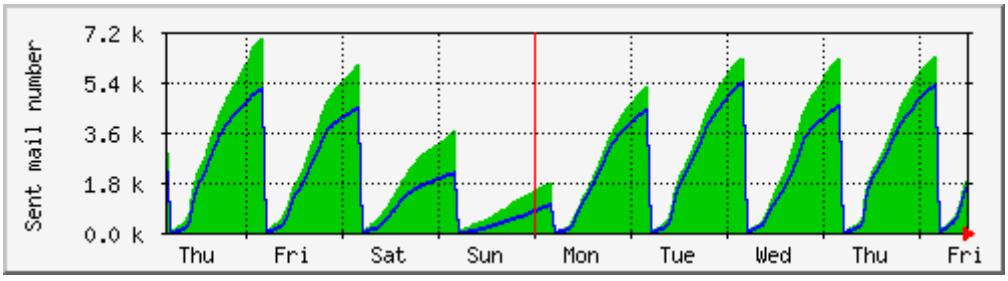
/etc/inetd.conf 에

smtp-stats stream tcp nowait nobody /usr/sbin/tcpd /usr/sbin/mailstats

와 같이 추가하면 된다.

위 설정이 끝난 후 5 분 간격으로 정기적으로 실행하여 주기 위해 /etc/crontab 파일을 열어 0-59/5 * * * * root run-parts /usr/local/mrtg/run/mrtg mail.cfg 와 같이 추가해 주면 매 5 분마다 자동으로 실행하여 아래와 같이 그래프를 생성하게 된다. 아래의 경우 하루 평균 약 5000 여 통이 메일이 전송되고 있는 것을 알 수 있다.

기타 Mrtg 의 사용방법에 대해 궁금하신 분은 본지 8 월호 part3 “시스템 모니터링과 자동화하기” 를 참고하거나 http://www.mrtg.org/ 또는 http://www.mrtg.co.kr/ 을 참고하기 바란다..



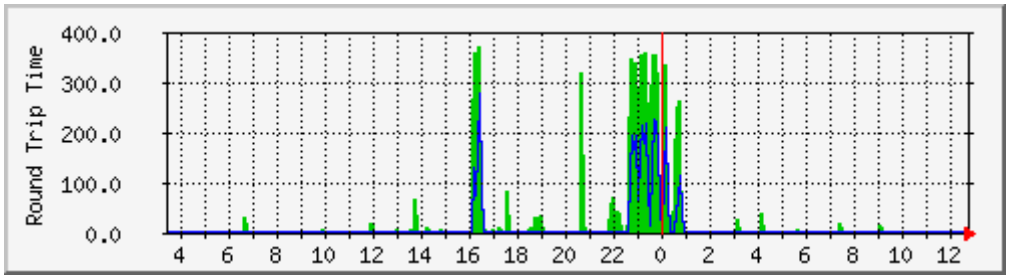
< 그림 7. 전송되는 메일 개수를 mrtg 로 본 화면>

mrtg 로 네트워크 속도 모니터링하기

추가적으로, mrtg 를 활용하여 특정 네트워크와의 접속 속도등을 모니터링하는 방법을 알아보도록 하자. 이를 위해서는 mrtg-ping-probe 라는 별도의 프로그램을 사용하면 되는데, 이 프로그램을 이용하면 특정 호스트 사이의 ping 소요 시간, 패킷 로스등을 그래프로 볼 수 있다. 이 프로그램은 http://pwo.de/projects/mrtg/ 에서 해당 파일을 다운로드 후 압축 해제하여 아래와 같이 mrtg 를 위한 cfg 파일을 만든 후 역시 mrtg 로 실행해 주면 된다.

```
Title[www.bora.net]: ping
MaxBytes[www.bora.net]: 5000
AbsMax[www.bora.net]: 10000
Target[www.bora.net]:`/usr/local/mrtg/mrtg-ping/mrtg-ping-probe www.bora.net'
YLegend[www.bora.net]: Round Trip Time
ShortLegend[www.bora.net]: ms
Legend1[www.bora.net]: Maximum Round Trip Time in Milli Second
Legend2[www.bora.net]: Minimum Round Trip Time in Milli Second
Legend3[www.bora.net]: Maximal 5 Minute Maximum Round Trip Time
Legend4[www.bora.net]: Maximal 5 Minute Minimum Round Trip Time
WithPeak[www.dacom.co.kr]: ymwd
```

위 설정은 실제로 필자의 네트워크에서 www.bora.net 과의 ping 결과를 모니터링하기 위한 설정으로 위와 같이 설정 후 /etc/crontab 에
0-59/5 * * * * root run-parts /usr/local/mrtg/run/mrtg ping.cfg 와 같이 설정해 주면 매 5 분마다 ping 소요 시간을 모니터링하게 되고 다른 ISP 와의 네트워크 속도는 Target 부분에서 www.bora.net 을 www.kornet.net 등과 같이 모니터링하고자 하는 ISP 로 적절히 변경하여 주면 된다.
위와 같이 설정후 mrtg 를 실행해 주면 아래와 같이 ping 으로 소요되는 시간을 그래프로 보여주게 된다. 아래의 경우 16 시와 오전 0 시를 전후로 네트워크에 장애가 있어 ping 수치가 갑자기 높아진 것을 알 수 있다.



< 그림 8. 모 ISP 와의 ping 결과를 mrtg 로 본 화면>

침입탐지 시스템(IDS) 운영하기

2001 년은 IDS(Intrusion Detection System) 의 해라고 해도 무방할 정도로 IDS 에 대한 많은 솔루션의 개발과 관련 정보 교류가 있었던 해였다. 리눅스에서는 SNORT 라는 공개 프로그램이 가장 유명하고 또 많이 사용되는 IDS 인데, snort 는 실시간으로 트래픽 분석과 패킷 로깅을 수행하는 네트워크 기반의 IDS(NIDS) 로 버퍼 오버플로우, 포트스캔, CGI, OS Fingerprinting 등의 룰을 이용하여 각종 접근 시도를 실시간으로 감지하는 기능을 가지고 있다. 또한 각종 플러그인 프로그램이나 응용 프로그램을 이용하여 확장된 이용이 가능하다.
SNORT 를 사용하려면 먼저 ftp://ftp.ee.lbl.gov/libpcap.tar.Z 에서 libpcap 라이브러리를 설치후 http://www.snort.org/ 에서 Snort 관련 파일을 다운로드 하여 아래와 같이
./configure
make
make install 로 설치하면 된다.
각종 룰은 설치시 기본적으로 제공되는 룰을 사용하여도 되고 홈페이지에서 지속적으로 갱신되는 rule 을 다운로드 받아 snort.conf 설정에 include 하여 추가 설정해도 된다.
그리고 Snort IDS 가 탐지하는 각종 로그 파일이 저장될 디렉토리인 /var/log/snort 를 생성한 후
./snort -d -l /var/log/snort -c snort.conf -A full -D 로 실행하면 실시간으로 트래픽을 분석하여 각종 침입탐지 룰에 해당하는 내용들이 로그 파일에 기록되게 된다.
tail –f /var/log/alert 를 보면 실시간으로 탐지가 되는 내용을 볼 수 있다.
그러나 이 로그 파일들이 너무 많고 또 복잡하여 일일이 분석하기가 어려우므로 쉽게 분석할 수 있도록 별도의 플러그인 프로그램을 이용할 수 있다.
http://www.snort.org/downloads.html 를 참고하면 많은 관련 프로그램들이 있으니 참고하기 바란다. 그 중 많이 사용되는 프로그램으로는 웹 기반에서 침입 탐지 분석이 가능한 Snortsnarf 라는 프로그램이 유명하다.
이 프로그램은 http://www.silicondefense.com/software/snortsnarf/ 에서 다운로드 가능하며 파일을 다운로드 후 압축을 해제하고 아래와 같이
cp -a include/* /usr/lib/perl5/site_perl/5.x.0/ 로 복사하고
cp cgi/* /usr/local/apache/cgi-bin/ 으로 복사한다. 그리고 Time-modules 디렉토리로 이동한

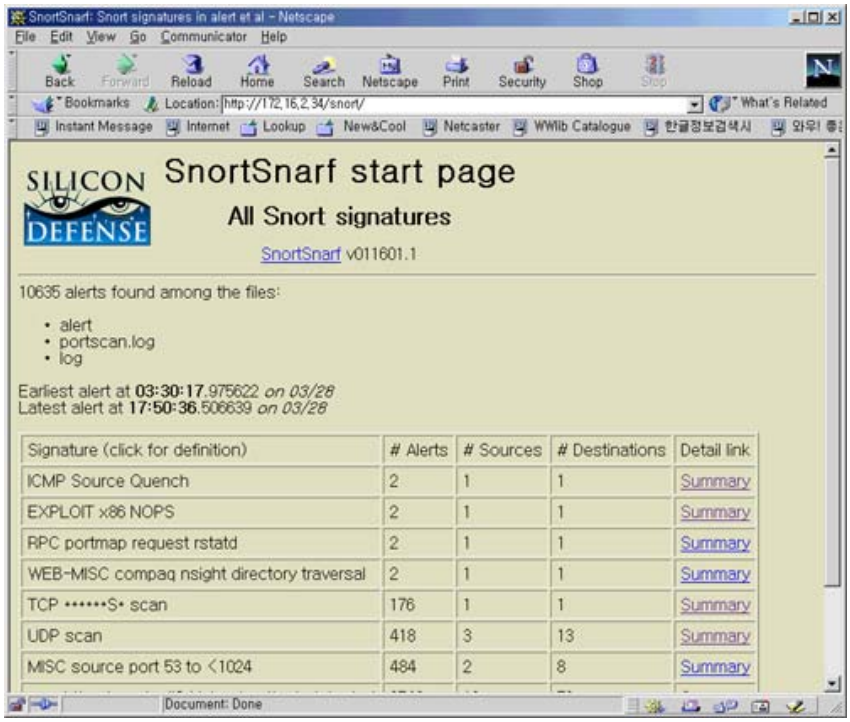
후 perl Makefile.PL ; make ; make test ; make install 로 Time 모듈을 설치한다.

실행은 ./snortsnarf.pl -rulesdir /usr/local/snort-1.8.1-RELEASE -rulesfile /usr/local/snort-1.8.1-RELEASE/snort.conf -d /usr/local/apache/htdocs/snort /var/log/snort/alert /var/log/snort/portscan.log /var/log/snort/log

를 start.sh 와 같은 한 줄의 스크립트 파일로 만들어 실행하면 로그 파일을 분석하여 분석 결과를 HTML 파일로 변환하여 준다. 이때의 실행 경로는 자신의 경로에 맞게 수정하면 된다. Snortsnarf 가 작동하는 실제 예제는

http://www.silicondefense.com/software/snortsnarf/example/index.html 에서 참고할 수 있다.

이 외 파일 무결성 체크, 네트워크 상태 점검, 침입탐지 시스템등의 기능을 통합하여 웹기반으로 한꺼번에 관리가 가능한 상용 NMS 나 IDS 수준의 DEMARC 라는 프로그램도 있다. 이 프로그램은 리눅스 및 기타 유닉스 계열에서 작동하며 Snort 1.8 이상 , Mysql 3.23 이상 기타 CGI, DBI, DBD::mySQL 등의 perl 모듈을 필요로 하며 이 프로그램에 대한 자세한 설치 방법 및 사용에 대해서는 http://www.demarc.org/ 를 참고하기 바란다.



< 그림 9. SnortSnarf 실행 화면 >

일반적으로 IDS 는 (1)방화벽 앞단, (2)방화벽 뒷단, (3)호스트 앞단등에 설치 가능한데, 통상적으로 방화벽(침입차단 시스템) 앞 단의 스위칭에서 모니터링하며 더미 허브일 경우에는 관계 없지만 스위치의 경우 별도의 미러링 포트를 통해 실시간으로 탐지하게 된다. 그러

므로 대용량의 트래픽 환경에서도 패킷을 잃지 않고 탐색하는지 여부가 IDS 의 성능에 크게 좌우를 하게 된다.
또한 호스트 기반의 IDS 에 비해 이러한 네트워크 기반의 IDS 는 모든 네트워크 트래픽을 탐지, 분석하여야 하므로 일정 정도의 네트워크 부하를 유발하게 되는 단점도 있다.

SMTP Relay 모니터링하기

스팸 메일은 어제 오늘의 문제가 아닐 정도로 사회적으로도 큰 문제가 되고 있다.
일반적으로 스팸 메일은 Relay 가 허용된 메일 서버를 이용하여 무차별적으로 보내어지는데, 이를 통해 스팸 메일을 보내느라 시스템에 불필요한 로드가 올라가고 많은 트래픽을 소모하게 된다. 최근에는 Relay 가 허용된 메일서버를 자동으로 검색하여 이를 알려주는 사이트가 생겨나기도 했는데, 자신이 관리하는 서버중에 Relay 가 허용된 메일 서버는 없는지 검색해 보도록 하자. 일반적으로 SMTP 로 사용하는 sendmail 의 경우 sendmail.cf 파일에서 Relay 여부를 설정하는데, 관리하는 서버가 많을 경우에는 일일이 확인해 볼 수 없으므로 아래와 같이 SMTP Relay 여부를 확인할 수 있는 스크립트를 사용할 것을 추천한다. 이 스크립트는 http://www.unicom.com/sw/rlytest/rlytest 에서 다운로드할 수 있으며 이 파일을 서버에 check.cgi 로 저장하여 **./check.cgi tt.co.kr** 과 같이 실행하면 해당 서버(tt.co.kr) 에서 Relay 가 허용되는지 여부를 알려준다.

예 1)
[root@www relay]# **./check.cgi tt.co.kr**
Connecting to tt.co.kr ...
<<< 220 www10.tt.co.kr ESMTP Today and Tomorrow(http://tt.co.kr/)
>>> HELO relay.tt.co.kr
<<< 250 www10.tt.co.kr Hello [211.47.65.15], pleased to meet you
>>> MAIL FROM:<nobody@tt.co.kr>
<<< 250 2.1.0 <nobody@tt.co.kr>... Sender ok
>>> RCPT TO:<root@relay.tt.co.kr>
<<< 550 5.7.1 <root@relay.tt.co.kr>... Relaying denied.
check.cgi: relay rejected - final response code 550

와와 같은 경우에는 Relay 가 허용되지 않는 경우(reject)이며

[root@www relay]# **./check.cgi mailserver.tt.co.kr**

Connecting to 210.34.6.193 ...
<<< 220 mailserver.tt.co.kr IMS SMTP Receiver Version 0.83 Ready
>>> HELO relay.tt.co.kr
<<< 250 OK
>>> MAIL FROM:<nobody@[210.34.6.193]>
<<< 250 nobody@[210.34.6.193] OK
>>> RCPT TO:<root@relay.tt.co.kr>
<<< 250 root@relay.tt.co.kr OK
>>> DATA
<<< 354 Ready for data
>>> (message body)
<<< 250 Message received OK
>>> QUIT
<<< 221 mailserver.tt.co.kr closing
check.cgi: relay accepted - final response code 221

위와 같은 경우는 Relay 가 허용된 경우(accept)임을 알 수 있다.
그런데, 일일이 모든 서버에 대해 위와 같이 확인해 보기도 귀찮고 하니 아래와 같이 일정 IP 대역의 서버를 모두 조회하여 Relay 가 허용되는 메일 서버가 확인되면 관리자에게 해당 서버에 대한 정보를 메일로 발송할 수 있도록 스크립트를 짜서 설정할 수 있다.

```
#!/usr/bin/perl

# check SMTP relaying or not between some IP ranges.

$TO_MAIL      = 'antihong@tt.co.kr';   //Relay 가 허용되는 서버 발견시 발송될 E-mail
$SUBJECT      = "[경고] Relay Allowed";  // 발송될 메일의 제목
$MAIL_PROGRAM  = "/usr/sbin/sendmail";

for($i=1;$i<=254;++$i){                  // Relay 를 조회할 IP 대역 끝자리
$TASKS[$i] = `./check.cgi 201.12.211.$i`;    // Relay 를 조회할 IP 대역
if ($TASKS[$i] =~ /accept/){
 &scan_print;
 }
}
```

```
sub scan_print {
open(MAIL, "|$MAIL_PROGRAM -t");
    print MAIL "To: $TO_MAIL ₩n";
    print MAIL "Subject: $SUBJECT ₩n";
    print MAIL "Content-type: text/html₩n₩n";
    print MAIL "<p>₩n";
    print MAIL "SMTP RELAY Allowed <b>201.12.211.$i</b> Server₩n";
close(MAIL);
}
```

위 스크립트를 relay.cgi 로 저장하고
http://www.unicom.com/sw/rlytest/rlytest 에서 다운받은 파일을 check.cgi 로 저장하여

두 개 파일을 같은 디렉토리에 두고 relay.cgi 를 각자의 IP 대역 환경에 맞도록 변경후 ./relay.cgi & 로 백그라운드로 실행하면 해당 IP 대역(위의 경우에는 201.12.211.0 대역)을 조회하여 Relay 가 허용된 서버를 발견시 서버의 IP 에 대한 정보를 지정한 E-mail 로 발송하게 된다. relay.cgi 에서는 조회할 IP 대역, 통보받을 e-mail 주소만 변경하여 사용하면 된다. 또한 서버의 환경이 자주 변경된다면 while 문을 사용하여 무한 루프로 작동하게 하거나 cron 에 설정하여 정기적으로 Relay 여부를 점검할 수 있도록 할 수도 있다.

만약 Relay 가 허용된 서버를 확인하였을 때는 /etc/mail/sendmail.cf
(또는 /etc/sendmail.cf) 파일을 열어

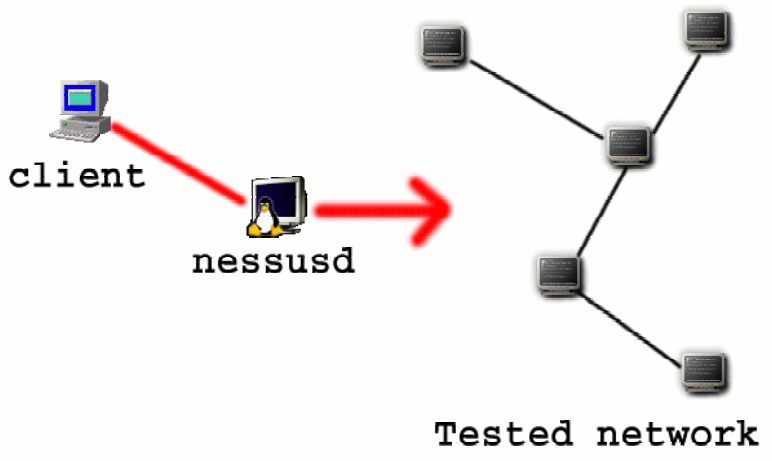
```
# anything else is bogus
R$*                     $#error $@ 5.7.1 $: "550 Relaying denied"
```

부분이 주석되어 있는지 확인한 후 주석이 되어 있으면 주석을 삭제한 후 sendmail 을 재가동하면 Relay 가 거부된다.

원격에서 보안 취약점 점검하기

보안에 조금이라도 관심있는 리눅서라면 SATAN 이나 SAINT 등의 프로그램 이름을 들어보았을 것이다. 이러한 종류의 프로그램은 원격에서 시스템이나 네트워크의 보안 취약점을 점검하기 위한 프로그램인데, 어떤 목적으로 사용되느냐에 따라 시스템 점검이 아니라 악용이 되어 시스템을 침해하기 위한 수단으로 사용되어 때때로 문제가 된 적도 있었다.
최근에 사용되는 취약점 점검 프로그램중에서는 원격지에서 쉽게 보안 취약점을 점검할 수 있는 프로그램으로 Nessus(http://www.nessus.org/) 를 추천한다. Nessus 는 컨설팅업체에서 보안 컨설팅을 하고자 할 때 사용하기도 하는 프로그램으로 nessus 에 대해서는 본지 9 월호에 “스캐너를 이용한 시스템 검사 2” 에서 자세히 설명되었으므로 간단히 소개만 하겠다.

Nessus 의 가장 큰 특징중 하나는 다른 프로그램과 달리 서버와 클라이언트 구조로 이루어졌다는 것이다. 서버에서는 600 여개의 각종 플러그인등이 설치되며 클라이언트에서 서버로 접속을 하여 정해진 룰에 따라 원격지의 취약점을 점검하는 것이다. 여러 방법이 가능하지만 여기에서는 리눅스에 nessus 서버를 설치하고 윈도우에 클라이언트 프로그램을 설치하여 취약점을 점검하는 법을 알아보도록 하겠다. (물론 리눅스에 서버와 클라이언트를 모두 설치해도 된다.)



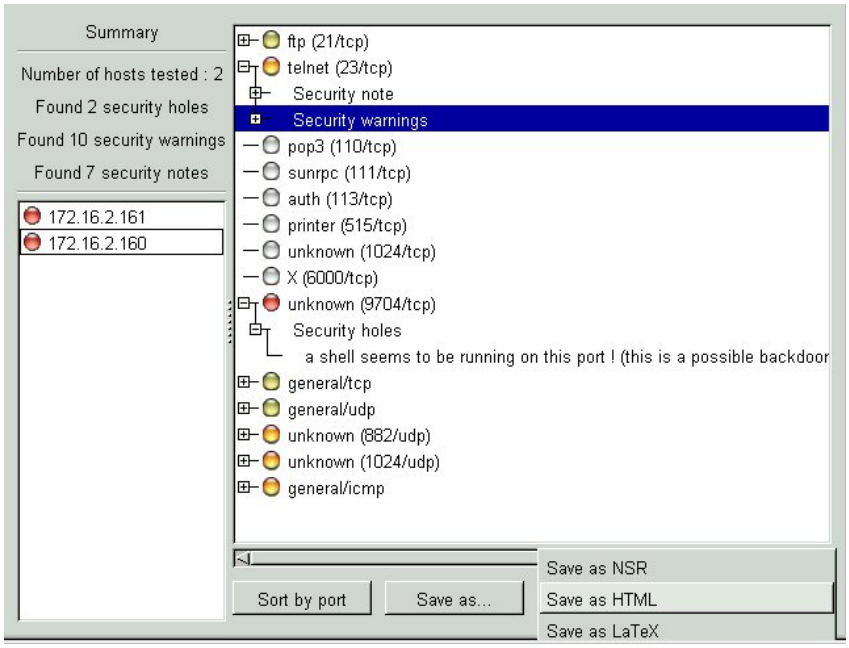
< 그림 10. Nessus 의 서버-클라이언트 구조>

Nessus 는 자체적으로 스캔을 위해 nmap 을 사용하므로 사전에 nmap 이 설치되어 있거나 없을 경우에는 http://www.insecure.org/nmap/ 에서 nmap 을 다운로드 하여 설치하여야 한다.
Nessus 서버나 클라이언트 설치법은 매우 쉽다. 단지 설치하고자 하는 서버에서
lynx -source http://install.nessus.org | sh 만 실행 후 묻는 질문에 적절히 답을 하거나 엔터만 치면 설치 및 셋팅이 완료된다.
(Are you behind a web proxy ? [y/n] 에서 프록시가 없을 경우 n 으로 답하면 다운로드 및 설치를 시작한다.) 다운로드가 끝나면 자동으로 시스템 체크 및 컴파일까지 수행하며 잠시 후 설치가 완료되면 /usr/local/sbin/ 에 관련 파일들이 설치된다.

/usr/local/sbin/nessus-adduser 를 실행 후
login : linuxwork
Authentication method : 입력 없이 엔터
Source host or network : nessusd 에 접속할 클라이언트 IP 또는 IP 대역을 입력한다.
여기에서는 취약점을 점검할 윈도우 프로그램을 설치할 PC 의 IP 나 IP 대역을 지정한다. (예: 192.168.1.3 또는 192.168.1.0/24)
One time password : 는 linuxwork 입력후 Ctrl-D 를 누르면 된다.
/usr/local/sbin/nessus-update-plugins // 플러그인 업데이트를 하고
/usr/local/sbin/nessusd -D // nessusd 를 데몬 모드로 실행하면

서버에서의 설정은 끝났다.

이제 http://www.nessus.org/win32.html 에서 윈도우용 클라이언트 프로그램을 받아 윈도우에 설치 후 bin 디렉토리에 있는 nessus.exe 를 실행하면 로그인 화면이 뜨는데, Port 와 Encrption 은 그대로 두고 Nessusd Host 에 데몬이 설치된 서버의 IP 와, Login 에 방금 추가 설정한 아이디로 로그인하면 된다.
이후 플러그인 메뉴에서 취약점을 점검할 내용을 선택한 후 Target selection 에서 스캔을 할 대상 서버의 IP 나 IP 대역을 설정 후 Start the scan 을 클릭하면 취약점 점검을 위한 스캐닝을 시작한다.
취약점 점검이 끝나면 모든 정보를 종합하여 어떤 부분이 취약한지와 또한 어떻게 패치하고 대처하여야 하는지에 대하여 결과에 대해 리포트를 작성해 보여주고 이 결과 보고서는 HTML 이나 ASCⅡ등 원하는 형태로 선택하여 저장할 수 있다. 자신의 서버에는 어떠한 취약점이 있는지 각자 nessus 로 점검을 해 보기 바란다.
추가적인 Nessus 이용 방법에 대해서는 http://nessus.org/ 를 참고하기 바란다.

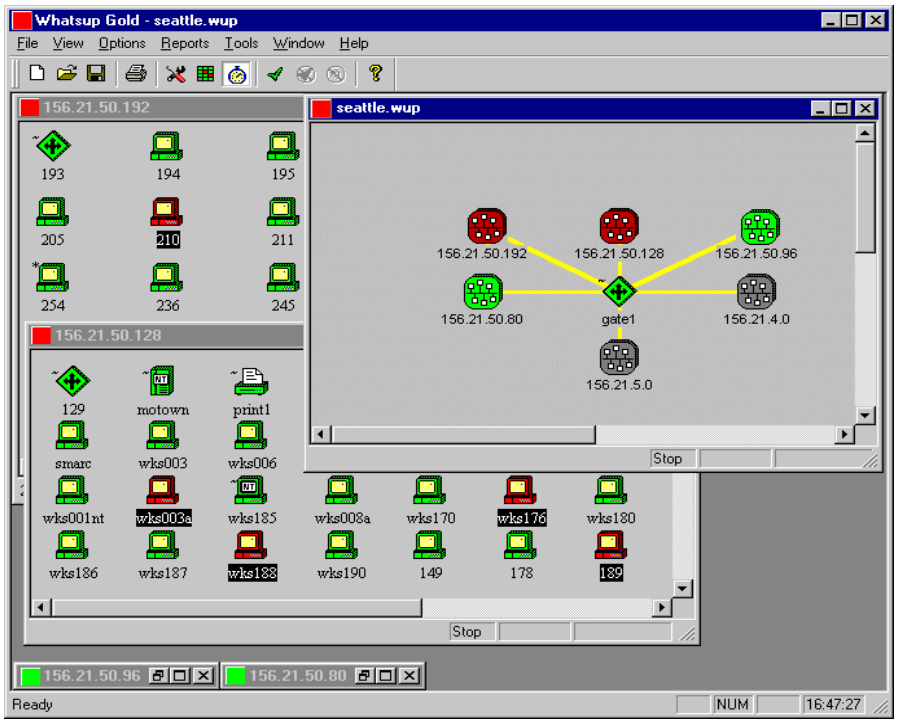


< 그림 11. Nessus 의 취약점 점검 결과>

시스템 및 네트워크 장애 모니터링

관리하는 서버가 많아지다 보면 특정 데몬이 작동하지 않거나 시스템이 다운되었을 경우 알 수 있는 방법이 없다. 심지어 출근해서 서버를 모니터링해 보면 퇴근 이후 줄곧 서버가 다운되어 있는 경우도 있는데, 고객으로부터 항의가 들어오고 나서야 확인이 가능하게 된다.

이를 위해 주기적이고 지속적으로 시스템을 모니터링하여 특정 서비스나 시스템이 다운시에는 다운 여부를 알려줄 수 있는 시스템 및 네트워크 모니터링 프로그램을 필요로 하게 된다. 일반적으로 이러한 프로그램은 NMS(Network Management System) 또는 SMS(System Management System) 라는 이름으로 매우 고가에 판매되고 있는데, 많은 기능을 제공하면서도 부담 없이 사용 가능한 프로그램으로 가장 추천할 만한 것으로는 와츠업(Whatsup)이라는 제품이 있다. 이 프로그램은 WS_FTP 를 제작한 Ipswitch 라는 업체에서 제작하여 배포하는 프로그램으로 유료 버전은 가격이 고가이지만(약 110 만원정도) 30 일 무료 체험 버전을 다운로드 받아 기능의 제한 없이 사용 가능하다. 쉬우면서도 강력한 기능으로 국내외 회선 업체나 IDC 등에서도 시스템이나 네트워크를 감시하기 위해 실제로 많이 사용되는 제품이다. 이 프로그램은
http://www.ipswitch.com/Products/WhatsUp/index.html나
http://solvnt1.solvithightech.co.kr/ipswitch/WhatsUp/ 에서 다운로드하여 실행하면 사용 가능하며 지정된 서버나 네트워크가 접속되지 않거나 특정 서비스(sendmail, http 등)가 작동하지 않을 경우에는 그림과 같이 해당 아이콘의 색이 변하면서 사운드 알람 , 비퍼, 페이저, 이메일, 음성 메시지로 해당 시스템이나 네트워크가 다운되었음을 통지하게 된다. 또한 로그 파일에 관련 정보를 저장하여 차후에 각종 통계를 위한 기능으로 사용할 수도 있다. 실제로 필자가 테스트해 보았을 경우에는 시스템이 다운 후 몇 초만에 다운 여부를 알려주었다. 이 프로그램의 한 가지 단점이라면 리눅스가 아닌 윈도우에서 작동하는 프로그램이라는 점이며 이와 비슷한 프로그램으로 리눅스에서 작동하는 프로그램을 원한다면 http://www.opennms.org/ 를 참고하기 바란다.



< 그림 12. 와츠업 프로그램 작동 예제>

라우터에서 네트워크 모니터링하기

리눅스와는 직접적으로 관련은 없지만 네트워크에서는 빠질 수 없는 장비가 바로 라우터이므로 간략히 소개만 하도록 하겠다. 라우터를 다룰 수 있는 환경이 된다면 직접 따라 해 보는 것도 좋지만 그렇지 않은 경우라면 참고만 하기 바란다.
네트워크를 크게 WAN 구간과 LAN 구간으로 나눈다면 WAN 구간의 통신은 라우터라는 장비를 통한 라우팅 프로토콜을 통해 이루어지게 된다. (참고로 LAN 구간에서는 IP 주소와 MAC 주소를 서로 매칭시켜주는 ARP 라는 프로토콜이 매우 중요하다.)
어차피 내부 네트워크가 외부 네트워크와 인터넷 통신을 하려면 라우터라는 장비를 반드시 거치게 되어 있으므로 모든 트래픽의 관문이라 할 수 있는 라우터에서의 네트워크 모니터링은 매우 효과적이며 효율적이다. 라우터에서 사용되는 명령어와 기능은 매우 많지만 가장 대표적인 기능중 몇 가지만 소개를 하겠다.
아래는 가장 많이 사용되는 Cisco 라우터에서 특정 회선의 패킷을 모니터링하는 IP Accounting 예제를 보여준다.

```
ROUTER#conf t                                  // config 모드로 변경
ROUTER(config)#int serial1                     // serial1번 라인 선택
ROUTER(config-if)#ip accounting                // ip accounting 기능 설정
ROUTER#show ip accounting                        // ip accounting 결과 보기
   Source            Destination          Packets              Bytes
 211.17.47.11      65.192.28.33                 2                120
 211.17.45.28      209.85.13.32                 1                 60
 211.17.45.184     216.32.243.7                 1                 44
 211.17.47.112     198.41.3.54                  5                212
 211.17.45.34      198.41.0.4                   2                 98
 211.17.46.4       209.247.164.36               1                112
 211.17.47.13      216.239.46.140              42              56431
 211.17.45.193     207.112.224.10              53              79122
 211.17.45.166     212.187.59.97               14              21000
 211.17.45.4       205.152.37.254               1                115
 211.17.45.187     216.87.103.212              19              28500
```

위의 IP Accounting 은 serial1 번 회선을 통해 라우팅되는 목적지 ip 와 소스 ip 그리고 각 패킷의 양과 사이즈등의 정보를 보여준다. 만약 특정 목적지(Destination) 나 소스(Source) 에서의 트래픽이 특별히 많을 경우에는 서비스 거부공격(DoS)이나 분산 서비스 거부 공격(DdoS) 여부를 의심해 보아야 한다.

아래는 netflow 설정 예제를 보여준다.

```
ROUTER#conf t                              // config  모드로 변경
ROUTER(config)#int serial1                 // serial1 번 라인 선택
ROUTER(config-if)#ip route-cache flow      // netflow 기능 설정
ROUTER#show ip cache flow                    // netflow 결과 보기
IP packet size distribution (360715 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .832 .036 .007 .007 .006 .013 .005 .004 .011 .008 .007 .009 .006 .004

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .004 .003 .007 .004 .017 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456704 bytes
  159 active, 65377 inactive, 50342 added
  789091 ager polls, 0 flow alloc failures
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet           1      0.0        10    55      0.0      71.6      15.6
TCP-FTP            916      0.0         3    47      0.0       3.9       8.8
TCP-FTPD            12      0.0         5   500      0.0       6.3       5.4
TCP-WWW          27967      0.0         9    92      0.0       7.0       4.6
TCP-SMTP          2555      0.0        11   221      0.0       7.8       4.4
TCP-other         1219      0.0         4   180      0.0       4.4       8.5
UDP-DNS          14424      0.0         2    64      0.0       6.7      13.8
UDP-NTP             22      0.0         1    76      0.0       0.0      13.1
UDP-other         4733      0.0         1   152      0.0       0.6      14.4
ICMP               714      0.0         7   300      0.0       8.1      14.6
Total:           52563      0.0         6   104      0.0       6.3       8.3
```

위와 같은 설정은 netflow 라 하는데, 위의 경우 serial1 번 회선을 통해 라우팅되는 패킷 사이즈별로 트래픽 상황과 각 프로토콜별로 분류된 상황을 모니터링할 수 있다.
이외 access-list 를 이용한 강력한 접근 제어 및 기타 여러 모니터링 방법이 있으니 관심 있는 분은 라우터 관련 서적이나 매뉴얼을 참고하여 구체적인 설정 방법을 익히기 바란다.