주 제 : 초보 데비안 사용자를 위한 네트워크 보안 이해

작성자 : 박병준 (BPark@computer.org)

작성일시 : 2001. 3. 10

All right reserved. copyright 2001. Byeong-Jun Park. 이 문서의 배포와 수정은 GPL 규정에 따릅니다.

요 약

리눅스에 처음 입문한 사람들에게 다중 사용자 환경(Multi-user)에 대한 체계적인 이해와 함께 시스템 보안의 관점에서 바라 본 네트워크의 기본 개념을 익히는데 중점을 두고 있습니다. 즉 리눅스를 처음 입문한 사람으로 관리자가 되기 위한 준비를 하는 과정에 있는 사람들에게 시스템 관리의 기본과 전체적인 이해를 도와주기 위한 것입니다.

먼저 사용자에 대한 이해 및 파일과 디렉토리의 권한 설정에 대한 이해와 이를 수행하는 방법을 공부합니다. 네트워크 서비스의 개념을 이해하고, 포트스캔 검사 도구 등을 이용하여, 간단히 자신의 시스템의 보안 상태를 확인할 수 있도록 배우게 됩니다

Ⅰ. 다중사용자(Multi-user)시스템의 이해

1. 강력해진 PC 어디에 쓰나

리눅스는 윈도우즈9x와 같은 운영체제와 본질적으로 다르다. 이는 마이크로소프트의 윈도우즈 운영체제가 개인용컴퓨터를 위한 운영체제로서 한 사람만이 사용한다는 가정하에 만들어진 반면, 리눅스(또는 유닉스)가 생각하는 컴퓨터는 동시에 여러 사람이 여러가지 일을 할 수 있도록 하는 것이기 때문이다. 여러가지 작업을 동시에 할 수 있는 환경을 보통 다중 작업, 또는 멀티태스킹(Multi-tasking)이라고 한다.

윈도우즈의 모체가 되는 DOS는 여러개의 작업을 동시에 할 수 없었다. 이것이 개선되어 윈도우즈에 와서는 동시에 여러개의 창을 띄워서 작업을 하는 멀티태스킹이 구현되었다. 리눅스는 이에 더해 한 사람이 여러개의 작업을 수행하는 것 뿐 아니라, 여러사람이 함께 사용하는 다중 사용자, 멀티 유저(Multi User) 환경을 지원한다.

2000년 초반에 200만원에 구입한 펜티움 III 550은 2년전에 약 천만원을 들여 구입한 썬 울트라 1 170 워크스테이션 (지금도 중고로 200만원은 넘을 것으로 예상) 보다 대략 네 배정도 빠른 연산 속도를 보여주었다. (썬 워크스테이션은 솔라리스7이 설치되어 있었다. 펜티움 III 머신에는 리눅스를 설치하였으며, 메모리 용량은 128메가로 동일하며, 프로그램은 수행중에 하드디스크에 접근하지 않는다. 똑같은 프로그램을 같은 버전의 GCC 컴파일러로 컴파일하여 수행한 시간을 비교하였다) 이는 이제 더 이상 나의 컴퓨터를 개인용으로만 한정시키기엔 컴퓨터의 연산 능력이 너무나도 뛰어남을 보여주는 것이다.

즉, 이전엔 외딴 초가삼간이던 개인 PC가 이제는 고속도로에 인접한 도심 속의 대저택이 된 것이다. 마이크로소프트 윈도우즈 같은 운영체제는 이 대저택을 나 혼자만의 놀이터로 만들어 준다. 그러나 경제적으로 볼 때 나 혼자 사는 대저택은 재미가 없다. 이 대저택을 리눅스로 재개발해서 아파트를 만들자. 위층엔 동생네, 아래층엔 부모님도 모시고 살자. 지금의 개인용 컴퓨터는 충분히 이만한 능력을 갖고 있다.

리눅스는 이렇게 훌쩍 성장해 버린 나의 강력한 PC에 날개가 되어주는 운영체제(Operating System)이다. 인터넷과 네트워크 환경에서 우리 가족이, 우리 부서가 모두 함께 사용할 수 있는 컴퓨터 환경을 리눅스는 제공해 주고 있다. 단 한가지 명심할게 있다. 외딴 초가삼간에선 문단속도 필요 없고, 혼자 떠들고 싶은 대로 떠들고 살아도 아무도 뭐라고 하지 않지만, 도심의 아파트에선 문단속을 하지 않으면 어떻게 되고, 혼자 시끄럽게 떠들면 어떻게 되는지 여러분들은 잘 알고 있을 것이다.

지금부터는 리눅스가 어떻게 함께 사는 일을 가능하게 하며, 다중 사용자 환경으로 인해 일어날 수 있는 여러 가지 보안 관련 문제점들을 어떻게 해결하고 있는지 배워보도록 한다.

2. 컴퓨터를 여럿이 함께 쓰기

집에 있는 PC를 가족 여러명이 같이 써 본 일이 있는가? 그렇다면 여러사람이 한 컴퓨터에서 문서 작업을 하고 전자우편을 사용했을 때, 일어나는 몇가지 고충을 경험한 적이 있을 것이다. 동생이 깔아놓은 게임에 걸려있던 컴퓨터 바이러스로 그동안 숙제해 놓은 것들 다 날리고, 윈도우즈를 다시 설치한 경험이 있거나, 처음 컴퓨터를 배우신 부모님들이 여기저기 연습으로 작성한 문서들을 못 찾아 헤 메고 있는 것을 본 경험이 한 번쯤은 있을 것이다.

금전적인 여유가 있다면, 컴퓨터를 또 사면되겠지 만, 대부분 서민층의 사람들은 한 대의 컴퓨터를 나눠 써야 하고 컴퓨터에 여러 사용자를 구분할 수 있는 능력과 각 사용자가 사용할 수 있는 영역을 제한시킬 수 있는 능력이 있으면 좋겠다는 생각을 해보았을 것이다. 또한 박학다식한 일부 사용자들은 윈도우즈 환경을 잘 설정하여 부분적이나마 이런 기능을 구현해서 사용하고 있기도 하다.

리눅스나 유닉스는 처음부터 이와 같이 한 대의 컴퓨터를 여러 명이 동시에 사용하는 것을 염두에 두었으므로, 여러 명의 사용자를 구분하고, 각 사용자가 접근할 수 있는 영역을 제한하고, 관리하는 여러 가지 방법을 고안하여 구현했다.

리눅스나 유닉스에서는 여러 명의 사용자에게 각각 이름이 되는 username (보통 흔히 ID라고 얘기하는 것)과 그에 상응하는 고유번호가 되는 uid를 부여하여 사용한다. 혼동의 소지가 있는 것은 uid는 우리가 보통 얘기하는 ID와는 다르게 마치 우리나라의 전 국민에게 부여하는 주민등록번호와 같은 개념의 것이다. 이것을 확인하는 명령어로는 id 라는 명령이 있다. 다음 예는 필자의 시스템에서 id명령을 실행한 결과이다.

```
bpark@fuzzy:~$ id
```

```
uid=105(bpark) gid=50(staff) groups=50(staff),6(disk),24(cdrom),29(audio)
```

여기서 필자의 username은 bpark 이며, uid는 105번임을 보여 준다. 리눅스에서는 이 uid 번호를 이용하여 각 사용자를 구분한다. 이는 마치 우리가 사용하는 주민등록번호와 같은 것이다. 역시 주민등록번호가 같은 사람이 있으면 안되듯이 리눅스에서도 같은 uid를 가진 다른 사용자는 없다.

여기서 앞의 예를 다시 보면 uid 뒤에 gid 라는 항목과 groups 라는 항목이 같이 있음을 알 수 있다. 이것은 리눅스나 유닉스에서 사용하는 group의 개념을 보여준다.

보통 사회에서는 같은 회사 동료들이 팀을 이뤄 어떤 일을 하게 된다. 다시 말하면 함께 일하면서 함께 공유하고, 작업하는 공동의 문서나 도구, 또한 결과물이 있게 마련이라는 뜻이다. 리눅스나 유닉스에서는 바로 이런 것을 위하여, user의 개념에 추가해서 group이라는 개념을 사용한다.

위의 예에서는 필자의 기본 소속 그룹이 staff 이며, 그 외에도 disk, cdrom, audio 그룹에 속해 있음을 알 수 있다.

이런 것이 필요한 이유는 바로 보안을 위해서이다. 어느 회사나 개인도 자신만이 볼 수 있는 일기와 같은 글이 있을 것이며, 내 동료나 가족만 볼 수 있는 글도 있게 마련이다. 이러한 자료나 문서에 대한 접근을 어떻게 제어할 것인가의 기준이 되는 것이 그룹과 user의 개념이 된다.

3. 리눅스 세계의 출생신고, 사망신고, 주민등록 ??

여기서는 리눅스의 로그인 관리 과정을 살펴보도록 하겠다. 리눅스를 시동하면 처음 부팅과 함께 /sbin/getty 이라는 프로그램이 실행된다. 필자의 데비안 머신에서 현재 수행중인 getty 프로그램을 살펴보기 위해서 ps 명령을 수행하였다.

```
bpark@fuzzy:~$ ps -ef |grep getty
root       502     1  0 Mar06 tty1     00:00:00 /sbin/getty 38400 tty1
root       503     1  0 Mar06 tty2     00:00:00 /sbin/getty 38400 tty2
root       504     1  0 Mar06 tty3     00:00:00 /sbin/getty 38400 tty3
root       505     1  0 Mar06 tty4     00:00:00 /sbin/getty 38400 tty4
root       506     1  0 Mar06 tty5     00:00:00 /sbin/getty 38400 tty5
root       507     1  0 Mar06 tty6     00:00:00 /sbin/getty 38400 tty6
bpark    18508 18504  0 03:14 pts/3    00:00:00 grep getty
```

이 명령은 현재 수행 중인 프로그램을 보는 ps 명령어에 -ef 옵션을 사용하여, 수행중인 모든 프로세스를 자세히 보여주도록 한 것이며 "|" 는 시프트 키와 역슬래시 키를 눌러서 넣는 것으로 "파이프" 라고 읽는다. grep 명령은 텍스트로부터 특정 패턴이 들어 있는 줄을 찾아 주는 명령으로 파이프는 ps명령의 결과를 grep명령의 입력으로 넣어주는 역할을 한다.

즉, ps -ef |grep getty 명령은 ps -ef 명령을 수행하여, 모든 프로세스를 출력한 것 중에 grep getty 명령으로 getty라는 문자열이 들어 있는 행만 보여 달라는 의미이다.

그렇다면 getty 프로그램은 어떤 일을 하는가? 로그인을 할 수 있는 경로를 감시한다. 자신이 감시하고 있는 연결이 활성화되면, /bin/login 이라는 프로그램을 실행시킨다. 그러면 우리가 리눅스를 로그인할 때 보이는 다음과 같은

```
Debian GNU/Linux testing/unstable fuzzy.hanyang.ac.kr
Access Restricted. Illegal login attempt is logged.

fuzzy login:
```

프롬프트를 띄우고, 로그인 이름을 받는 역할을 한다. 이 로그인 이름에 자신의 username을 적고 암호를 입력하면, 이제는 username과 암호를 검사하고, 시스템에 접근이 가능하도록 해 주는 중요한 역할을 한다.

이 때, login 프로그램은 /etc/passwd 라는 파일과 /etc/group 이라는 파일, 그리고 /etc/shadow 파일을 이용하여 검사를 수행한다. 지금부터 시스템 로그인과 사용자 관리에 핵심적인 역할을 하는 이 세 파일을 자세히 알아본다.

먼저 /etc/passwd 파일을 살펴보자. 이 파일은 리눅스, 유닉스 머신에서 주민등록부와 같은 역할을 하는 파일이다. 필자의 /etc/passwd 파일 일부를 살펴보자.

```
bpark@fuzzy:~$ more /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
 .... 중간 생략 ....
bpark:x:105:50:Byeong-Jun Park, Ph. D. Candidate:/home/bpark:/bin/bash
mysql:x:65533:65533:MySQL Server:/var/lib/mysql:/bin/false
```

이 내용의 각 줄은

계정:패스워드:UID:GID:GECOS:디렉토리:쉘

과 같은 문법 구조를 갖고 있다.

계정은 자신의 username을 의미하는 것이다. 패스워드 자리에는 모두 "x"로 채워져 있는데 이것은 실제 암호의 내용이 /etc/shadow 파일에 있다는 것을 의미하는 것이다.

UID와 GID는 앞에서 설명한 각 사용자와 그룹에 대한 고유번호를 의미하는 것이다. GECOS 항목은 해당 유저에 대한 설명으로 별 의미가 없이 쓰이기도 하고, 전자우편을 사용할 때, 보내는 사람 이름으로 사용되기도 한다. 디렉토리 항목에는 사용자의 홈디렉토리를 적어주는 것으로 홈디렉토리의 의미는 다음 장에서 더 자세히 설명하도록 하겠다.

쉘은 사용자의 명령을 해석하고, 그 결과를 사용자에게 보여주는 일종의 명령해석기 프로그램으로, 예전에 DOS를 사용해 본 사람은 command.com 이라는 프로그램과 같은 역할을 하는 프로그램으로 생각하면 된다. 이 쉘을 기록하는 자리에 예에서 보여주는 마지막 행의 mysql유저처럼 /bin/false 라고 적어 두면, 이 username을 이용해서는 로그인이 불가능하게 된다.

이제 /etc/shadow 파일을 보도록 하자.

```
bpark@fuzzy:~$ su
Password:
root@fuzzy:/home/bpark# more /etc/shadow
root:$1$gebSsEFW$Fx1nWWioNeUbibd8KvS90/:11327:0:99999:7:::
daemon:*:11312:0:99999:7:::
bin:*:11312:0:99999:7:::
sys:*:11312:0:99999:7:::
.....이하생략 ....
```

살펴보면 알겠지만 여기서도 역시 각 항목을 구분하는 글자가 ":" 이다. 맨 첫 번째 항목은 /etc/passwd 파일에서도 본, username이며, 두 번째 항목이 암호 항목이다. 이 길이가 상당히 긴 것을 알 수 있다. 그 뒤의 숫자들은 계정이 만들어진 시간, 암호를 언제 변경해야 하는지, 이 계정을 언제까지 사용할 수 있는지 등에 대한 기록으로 그냥 ":" 만 써 두면, 이런 제약을 사용하지 않는 것을 의미한다. 이 부분에 대한 자세한 내용은 man shadow 해보면 나오므로 참고하도록 하고, 여기서는 암호 부분에 대해 자세히 살펴본다.

암호 부분을 살펴보면 root에 대한 암호가 "1gebSsEFW$Fx1nWWioNeUbibd8KvS90/" 로 되어 있다. 이것은 필자가 사용한 키 입력을 md5 라는 해쉬 알고리즘을 이용하여 코드화[1] 한 것으로 해쉬 알고리즘은 역방향의 연산이 불가능한 특징을 갖고 있다. 즉, 이 값을 알고 있어도 역추적으로 처음 입력한 암호를 알 수 없음을 의미한다. 그러므로, 암호를 확인할 때는 사용자가 입력한 암호를 md5 알고리즘에 적용하여 그 결과 값과 /etc/shadow 파일에 저장된 값을 비교하여 두 값이 정확히 일치하면 정당한 암호로 인정하는 방법을 사용한다.

그러므로 암호를 해독하기 위해서는 임의의 입력을 md5 알고리즘 수행시켜서 그 결과가 같아질 때까지 반복하는 방법을 사용한다. 이런 것을 보통 패스워드 크래킹이라고 하며 아주 고전적이고 초보적인 방법으로 사용되는 해킹 방법이다.[2]

---

1) **MD5알고리즘으로 코드화**: 암호를 코드화하는 데는 MD5알고리즘이 현재 가장 좋은 방법으로 사용되고 있다. UNIX와 호환을 시키기 위해서는 MD5대신에 crypt라고 하는 고전적인 방법이 있다. 대부분의 리눅스를 처음 설치할 때, MD5와 crypt 중에 어떤 것을 사용할 것인지 물어보는데, UNIX와 함께 계정 정보를 공유하는 경우(이런 것을 보통 NIS라고 부른다)가 아니라면, MD5사용을 선택하는 것이 좋다.

2) **패스워드 크래킹**: 첩보영화나 SF영화에서 숫자나 알파벳이 반복되면서 암호를 맞춰서 문을 열거나, 파일을 열어보는 것이 이런 종류의 해킹에 해당된다. crack 과 같은 프로그램이 이런 일을 수행하며 해킹의 초보 단계로 많이 사용된다. 하지만 사실 이 프로그램은 관리자에게 암호가 너무 쉽게 설정해 놓은 사용자가 있지는 않은지 검사하는 도구로 사용되기도 한다. 93년에 발생했던 청와대 사칭 해킹 사건의 경우 청와대의 천리안계정 암호가 너무 쉽게 유추할 수 있는 숫자로 되어 있었기 때문에 발생한 경우에 해당된다.

필자가 위의 예에서 /etc/passwd 파일을 볼 때는, 그냥 보았지만, /etc/shadow 파일을 볼 때는 su 라는 명령을 먼저 수행하였다. 이 내용을 기억하고 있다가 다음 장에서 다시 살펴보도록 하겠다.

마지막으로 /etc/group 파일을 살펴보면

```
bpark@fuzzy:~$ more /etc/group
root:x:0:root,bpark
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:bpark
....이하생략....
```

group의 이름:x:GID:콤마로 구분된 그룹에 속한 username의 리스트
의 구조를 갖고 있다. 앞에 나온 id명령의 결과에 나온 groups 항목이 바로 이 파일에 의해 결정되는 것이다.

즉, 이 세 파일 /etc/passwd, /etc/shadow, /etc/group 들이 리눅스 세계의 주민등록부 같은 것으로 새로운 유저를 이 곳에 기록하고, 유저 확인을 위해, 이 파일들을 사용한다는 것을 알 수 있다. 물론 유저가 없어지면, 이 파일에서 반드시 해당 user 항목을 삭제해야 한다는 것도 의미하는 것이다.

II. 리눅스의 파일 구조와 파일 권한

1. 파일이란

파일(File)하면 대부분의 사용자는 데이터가 저장되거나 프로그램이 기록되는 정보단위 정도로 알고 있을 것이다. 그러나 이것은 파일의 한 일부분만을 알고 있는 것이다. 대부분의 운영체제에서는 파일구조가 아주 중요한 부분을 차지하며, 파일 입출력은 단순히 디스크에 자료를 기록하고 읽어오기 위한 시스템이 아니다.

리눅스나 유닉스에서는 디스크 상에 데이터가 저장되는 구조로 뿐만 아니라, 하드웨어 장치에 접근하기 위한 디바이스 드라이버가 모두 파일 구조로 만들어져 있다. 즉, 그래픽 화면 제어를 위해 비디오 장치와 통신을 하는 것도 파일에 저장하고 파일로부터 읽어오는 방식을 사용하며, MP3를 재생하는 것도, MP3를 음파로 해독한 다음, 이 데이터를 /dev/dsp 라는 파일에 기록한다. /dev/dsp는 바로 사운드카드를 지칭하는 파일인 것이다.

심지어는 현재 실행되고 있는 프로그램의 실행 환경을 저장하고 있는 메모리를 마치 디스크처럼 보이게 만들어 이러한 메모리 내용을 파일을 읽고 쓰는 방식으로 접근할 수 있도록 하고 있다. 이것이 /proc 파티션의 정체이다.

그러므로 파일구조를 이해하는 것은 리눅스가 동작하는 전반을 이해하는 데, 핵심적인 연결고리가 된다.

2. 리눅스 세상의 모든 것엔 주인이 있다.

리눅스의 파일 세계에는 모두가 주인이 있다. ls -l 명령으로 확인해 보자.

```
root@fuzzy:/# cd /home
root@fuzzy:/home# ls -l
합계 14
drwxr-xr-x    7 root     alumni        143  1월  5 13:14 alumni
drwxr-xr-x   36 bpark    root         2128  3월 10 02:27 bpark       *
drwxr-xr-x    2 root     root           35  1월  8 01:17 lost+found
drwxr-xr-x    6 mysql    mysql         126  2월 14 16:35 mysql-data
drwxr-xr-x    3 root     others         55  3월  9 16:51 others
....이하 생략....
```

* 표시가 된 필자의 디렉토리를 보자. 맨 처음의 drwxr-xr-x 항목은 다음 절에서 더 자세히 설명하겠지만 권한 설정에 대한 내용이며, 두 번째로 나온 36 이라는 숫자는 이 파일의 하드링크 번호라는 것으로, 디스크 상의 위치를 표시하는 고유번호 쯤으로 생각하면 된다. 세 번째 항목에 나오는 bpark 이 바로 이 파일의 주인의 username이다. 그 다음 root는 이 파일이 속한 group이며, 맨 마지막 bpark은 이 파일의 이름이다.

다시 설명하자면, bpark이라는 디렉토리(맨 마지막 항목)가 bpark이라는 유저(세 번째 항목)를 주인으로 가지며, root그룹에 속해 있고, 2128 바이트의 크기를 갖고 있다는 뜻이다.

앞장에서 홈 디렉토리라는 내용이 나왔었다. 바로 지금 위의 예에서 보고 있는 bpark이라는 디렉토리는 /home/bpark 이며, 이 디렉토리는 필자의 id인 bpark 유저의 홈 디렉토리이다. 앞장에 나온 /etc/passwd파일의 내용 중에

```
bpark:x:105:50:Byeong-Jun Park, Ph. D. Candidate:/home/bpark:/bin/bash
```

밑줄 친 부분에 해당되는 곳이 홈 디렉토리의 위치이며, 이 홈디렉토리의 주인은 바로 bpark이라는 유저이다.

홈 디렉토리는 각각의 유저가 자신의 작업공간으로 사용하며, 자신의 자료를 저장해 두는 자기 방과 같은 의미이다. 꼭 홈디렉토리 뿐만 아니라, 리눅스 공간 상에 있는 모든 파일에는 이렇게 주인과 그룹이 있다. 심지어는 /dev 디렉토리에 있는 장치 파일들(하드웨어를 직접 파일구조로 만들어서 접근할 수 있도록 해주는 것)조차도 주인과 그룹이 있어서, 이 하

드웨어를 사용할 수 있는 사람을 제한시켜 놓았다.

3. 리눅스 세계의 독재자

리눅스의 모든 파일에는 주인이 있다고 했는데, 그럼 이 주인을 결정하는 것은 누구일까? 바로 리눅스에서는 root 라는 유저가 이런 권한을 갖고 있다. 우리 사회로 치면, 최고 권력 기관인 셈이다.

파일의 주인을 새로이 설정하거나, 주인을 바꾸는 일은 이 root만이 할 수 있다. 우리 사회로 치면, 소유권 이전을 위해서, 등기소(법원)를 반드시 찾아야 하는 것이나 비슷하다고 하겠다. 한가지 차이가 난다면, 우리 사회에서는 양자의 합의 사실을 확인하기만 하면 소유권 이전이 되지만, 리눅스의 경우에는 이런 일을 양자의 합의와 관계없이 root가 해야 한다는 차이라고 하겠다.

그렇다면 왜 일반 사용자가 당사자끼리 합의에 따라 서로 소유권 이전이 가능하게 하지 않았을까? 그것은 바로 보안 문제와 관련이 있다. 만약 나쁜 생각을 가진 일반 사용자가 로그인 과정에서 암호를 확인하는 프로그램을 자기가 새롭게 작성한 후, 이것을 root 소유인 것처럼 바꿔 놓거나 한다면 심각한 문제가 아닐 수 없다.

그러므로 리눅스에서는 반드시 사용자가 생성한 파일은 해당 사용자가 주인이 되며, 이 주인 설정을 개인 사용자가 자의적으로 바꿀 수 없게 해 놓은 것이다.

파일의 주인을 바꾸는 명령은 다음과 같은 chown 명령을 사용한다.

chown [-R] *new_owner* *files.....*

뒤에 적힌 여러 개의 파일들의 주인을 new_owner로 바꾸는 명령이다. 만약, 여기에 -R 옵션을 붙이게 되면, 뒤에 나오는 files 영역 중에 디렉토리가 있으면, 그 안에 있는 모든 파일들에 대해 재귀적으로 chown을 수행하게 된다.

다음은 웹서버만이 사용할 수 있도록 test 디렉토리의 소유를 바꾸는 예이다.

```
bpark@fuzzy:~/test$ su
Password:
bpark@fuzzy:~/test$ ls -l
합계 1
drwxr-xr-x    2 bpark    staff          35  3월 10 18:13 test
root@fuzzy:/home/bpark/test# chown www-data test
root@fuzzy:/home/bpark/test# ls -l
합계 1
drwxr-xr-x    2 www-data staff          35  3월 10 18:13 test
```

이렇게 하면 test라는 디렉토리의 주인은 www-data가 되며, www-data는 웹서버 유저이므로 웹서버가 이 디렉토리의 주인이 된다.

그룹을 바꾸는 명령은 chgrp 이며, 문법과 구조는 chown과 같다. 물론 chgrp 명령도 root 유저 만이 할 수 있다.

만약 한 명령으로 소유주와 그룹을 모두 바꾸고 싶을 때는 chown명령에 유저와 그룹을 함께 써주면 된다. 다음 예는 앞의 test 디렉토리를 bpark의 소유로 바꾸고 그 그룹을 www-data[3]로 설정하는 예이다. 소유주의 username과 그룹이름 사이에는 "."을 찍는다.

```
root@fuzzy:/home/bpark/test# chown bpark.www-data test
root@fuzzy:/home/bpark/test# ls -l
합계 1
drwxr-xr-x    2 bpark    www-data       35  3월 10 18:13 test
```

또 한가지 유의할 것은 root 유저는 반드시 UID를 0번을 갖는다는 사실이다. 앞장에서 UID가 마치 주민등록번호와 같은 것이라고 설명하였다. 그렇다면, 같은 주민등록번호에 이름이 두 개가 있을 수 있을까? 답은 있다이다. 우리 사회에도 예명이라는 게 있듯이 유닉스 세계에도 같은 UID에 서로 다른 이름 두 개가 있을 수 있다. 만약 cracker 라는 유저의 UID가 0번이라면, cracker와 root는 같은 유저로 인식된다. 즉, 가짜 최고통수권자가 나타난 것이다. 그러므로 가끔 /etc/passwd 파일을 살펴보고, 혹시라도 UID항목이 0이 붙어 있는 사용자가 있지는 않는지 살펴볼 필요가 있다.

4. 내 것도 아닌데 볼 수 있을까?

우리는 앞에서 파일의 주인과 그룹에 대해 설명을 했다. 여기서는 이런 그룹과 사용자 그리고 같은 그룹에 있지 않은 사용자들에 대해, 나의 파일을 얼마만큼 공개할 것인지를 결정하는 문제가 있다. 이러한 일을 권한 설정(Permission) 이라고 한다.

모든 파일은 권한을 세 단계로 나눈다. 주인(Owner,User)에 대한 권한과 그룹(Group)에 대한 권한, 그리고 그 외 사람들(Others)에 대한 권한이다. 앞에서부터 계속 보아온 ls -l의 결과를 다시 한 번 보면,

```
drwxr-xr-x    2 bpark    www-data       35  3월 10 18:13 test
```

맨 앞에 drwxr-xr-x 항목이 있다. 여기를 네 곳으로 구분할 수 있는데, 맨 앞 한 글자와 나머지 아홉자를 세자씩 나누면 된다. 다음과 같다.

```
d   rwx   r-x   r-x
```

맨 앞 글자는 해당 파일의 특징을 나타내는 글자로 "-" 이 써져 있으면, 일반적인 파일을 뜻하며, "d"가 써져 있을 때는 디렉토리를 의미한다. 이 외에도 링크파일임을 표시하는 "l" 자가 오기도 하고, 장치 파일과 같은 특수 파일에 대해서는 "c" 나 "p"가 붙기도 한다.

---

3) www-data : 데비안에서 웹서버의 소유주와 그룹으로 사용되는 이름. 같은 www-data 라고 여기서는 그룹이름으로 쓴 것이다.

두 번째부터 세 개로 나눠지는 영역은 첫 번째가 주인(Owner,User)에 대한 접근 권한이며, 두 번째 영역이 group에 대한 권한이고, 마지막이 그 외 사람들에 대한 접근 권한이다.

여기에는 r,w,x,s,-,t 등의 항목이 있다. 이 중에서 먼저 위의 예에서 나온 rwx에 대해서만 먼저 알아본다.

r : 읽기 가능 (readable)
w : 쓰기 가능 (writable)
x : 실행 가능 (executable)

즉, 위의 예에서 test 라는 파일은 디렉토리를 저장하는 파일이며, 주인인 bpark 유저에게는 읽고, 쓰고, 그 안에서 프로그램을 실행할 권한이 있는 것이며, bpark이 아닌 다른 그룹내의 유저나 그룹원이 아닌 다른 사람들에게는 그 내용을 읽고, 그 안에서 프로그램을 실행할 수 있는 권한만 준 것이다.

이런 권한 설정을 변경하기 위해서는 chmod 명령을 사용한다.

chmod [-R] *mode files.....*

의 형식을 따른다. -R옵션은 앞에서 설명한 chown, chgrp 명령과 마찬가지로, 하위디렉토리까지 모두 이 설정을 적용하도록 하는 옵션이다.

mode의 자리에 오는 형식은 두 가지 형식이 있다. 한가지는 기호를 사용하는 것이고, 다른 한가지 방법은 8진수 숫자를 사용하는 것이다. 먼저 기호를 사용하는 법을 살펴본다. 권한을 바꿀 대상을 설정하는 알파벳과. 이의 설정을 추가할 것인가 제거할 것인가를 설정하는 +/- 기호, 그리고 마지막으로 어떤 권한 설정을 추가하거나 제거할 것인가를 설정하는 알파벳으로 되어 있다. 또한 이런 세 글자의 설정을 콤마 단위로 끊어서 여러 개를 쓸 수 있다.

[ugoa...][+-=][rwxXs]

u 는 user를 의미하는 것으로 이 파일의 주인에 대한 설정을 의미한다. g는 group을 의미하는 것으로 그룹에 대한 권한 설정이다. o 는 ohters의 기호로서 동일 그룹 외의 모든 사용자에 대한 권한이다. a는 all의 의미로 u와 g와 o를 모두 포함하는 것이다.

다음은 현재 디렉토리와 모든 하위 디렉토리에서 그룹과 그룹 외 사람들에 대한 쓰기 권한을 제거하는 예이다.

```
$ chmod go-w .
```

8진수를 사용하는 방법은 4,2,1 의 숫자의 조합을 사용하는데, 4는 읽기 권한(r), 2는 쓰기 권한(w), 1은 실행 권한(x)를 의미한다. 이것을 세 자리를 쓴다. 첫 번째 숫자는 user에 대한 권한이며, 두 번째 숫자는 group에 대한 권한이다. 마지막 숫자는 others에 대한 권한이다. 즉, 모든 권한을 직접 숫자로 적어주는 것이다.

이 숫자의 조합을 rwx기호와 연관해서 표시하면 다음과 같다.

```
rwx = 4+2+1 = 7
r-x = 4+1   = 5
r-- =       = 4
rw- = 4+2   = 6
--x =       = 1
-w- =       = 2
```

그러므로 앞에서 보여준 test디렉토리의 권한인 rwxr-xr-x는 숫자로 쓰면 755가 된다.

권한 설정의 한가지 예로 외부로부터 안전하게 보호하면서 웹서버의 PHP나 CGI만이 이 특정 디렉토리에 접근하도록 하고 싶을 때에는 해당 디렉토리의 그룹을 www-data로 만들고 그 디렉토리에 대해서 그룹에 대해서만 쓰기 권한을 설정해 준다.

```
root@fuzzy:/home/bpark/test# chown bpark.www-data test
root@fuzzy:/home/bpark/test# ls -l
합계 1
drwxr-xr-x    2 bpark    www-data       35  3월 10 18:13 test
root@fuzzy:/home/bpark/test# chmod g+w test
root@fuzzy:/home/bpark/test# ls -l
합계 1
drwxrwxr-x    2 bpark    www-data       35  3월 10 18:13 test
```

이제 test 디렉토리는 소유주인 bpark과 게시판이나 자료실 프로그램이 읽고 쓸 수가 있는 디렉토리가 되었으며, 그 외의 사람들은 이 디렉토리를 변경하지 못하게 해 준 것이다. 대부분의 게시판 CGI나 PHP프로그램에서는 이 권한을 777 즉, rwxrwxrwx로 설정하도록 설명해 놓았는데, 이는 매우 좋지 않은 방법이다.

5. 사용자를 변경하여 실행하기, 사용자 권한을 유지하여 보존하기

앞 절에 오기까지 일반 사용자인 bpark유저가 root권한을 얻을 때, su[4] 라는 명령을 사용하였다. 실험 예를 보자.

```
bpark@fuzzy:~$ id
uid=105(bpark) gid=50(staff) groups=50(staff),6(disk),24(cdrom),29(audio)
bpark@fuzzy:~$ su
Password:
root@fuzzy:/home/bpark# id
uid=0(root) gid=0(root) groups=0(root)
```

사용자의 구분은 리눅스에서 UID를 이용하여 한다고 말했는데, 결국은 su라는 명령이 bpark사용자의 UID를 105번에서 0번으로 바꿔 놓았다. 앞에서도 설명했지만 이런 일은 분명 root만이 할 수 있다고 했다. 그렇다면 su 명령에는 무슨 마법이 있는 것일까?

---

4) su : switch user 의 약자. 많은 사람들이 root로 권한을 변경할 때 사용하기 때문에 super user의 약자로 알고 있지만 이는 틀린 것으로, 시스템에 로그인한 사용자가 자신의 로그인 이름을 다른 사용자로 바꿀 때 사용하는 명령이다. 단, 이 명령의 인자로 아무것도 전달이 되지 않으면 su root 한 것과 같은 효과를 얻도록 되어 있을 뿐이다.

먼저 su 프로그램의 권한을 살펴보자.

```
bpark@fuzzy:~$ cd /bin
bpark@fuzzy:/bin$ ls -l su
-rwsr-xr-x    1 root     root        23100  1월 18 04:51 su
```

여기서 밑줄 친 s를 보면 앞 절에서 설명한 것과 다른 기호를 갖고 있다. su 라는 프로그램은 사용자의 로그인 이름을 변경하여 주는 것으로 누가 이 프로그램을 실행하더라고 마치 root가 실행하는 것과 같은 효과를 얻게 해주는 결과가 된다. 그러므로 bpark유저가 su를 실행하더라도 su프로그램은 root의 권한을 갖고 실행이 되는 것이다.

이런 것을 SetUID 라고 한다. 이 SetUID가 설정되어 있는 프로그램은 그렇게 많지 않다. 대부분은 일반사용자가 실행할 수 있지만, 시스템의 설정을 건드리거나 시스템 파일을 사용하는 경우에 해당되며, su, sendmail, passwd(명령) 등의 프로그램 등이 바로 이런 프로그램들이다.

이 SetUID도 chmod명령을 사용하여 설정할 수 있다. 이것은 8진수 숫자 네자리를 써서 설정하게 되는데, 뒷 세자리는 앞 절에서 설명한 권한 설정과 동일하며, 맨 앞의 숫자에 4를 써주면, 이 SetUID권한이 설정된다. SetUID권한을 없애고 싶을 때는 4대신에 0을 써주면 된다.

```
bpark@fuzzy:~/test$ chmod 4755 ttt
bpark@fuzzy:~/test$ ls -l ttt
-rwsr-xr-x    1 bpark    staff           0  3월 11 10:53 ttt
bpark@fuzzy:~/test$ chmod 0755 ttt
bpark@fuzzy:~/test$ ls -l ttt
-rwxr-xr-x    1 bpark    staff           0  3월 11 10:53 ttt
```

SetUID가 붙어 있는 프로그램들은 각별한 주의가 필요하다. 이 프로그램들이 외부의 경쟁조건(race condition)에 의해 비정상적으로 중단이 되었을 때, root 권한의 쉘을 내주는 경우가 있기 때문이다. 그러므로 이런 프로그램들은 버그가 발견되었을 때, 재빨리 버그가 없는 버전으로 바꿔주는 노력이 필요하다.

또한 시스템의 침입자가 시스템의 다른 버그를 이용하여 자신만이 실행할 수 있는 경로에 이런 SetUID가 설정된 프로그램을 심어 놓기도 한다. 그러면 이 침입자는 외부로부터 이 시스템 안에서 root의 권한으로 프로그램을 수행할 수 있게 된다.

III. 인터넷 서비스와 포트(port)

1. 실생활에 비유해 배워보는 인터넷 서비스

인터넷을 보통 정보의 바다라고 이야기한다. 지금부터는 바다와 배, 그리고 항구에 비유하여 인터넷 서비스와 서버의 개념을 이해해 보도록 하겠다.

사람들이 필요한 정보를 얻기 위하여 정보의 바다를 항해하듯이, 실제 화물선은 필요한 재화를 다른 곳에서 얻기 위하여 대양을 횡단한다. 대양을 횡단한 배가 바다건너 나라에 도착하면, 그 나라에서는 이 배가 들어올 수 있는 항구를 준비한다. 또한 이 항구에는 똑같이 생긴 부두만 있는 것이 아니라, 여러 종류의 접안 시설을 갖추어 놓았을 것이다.

사람이 내리는 여객선 터미널, 또 곡물을 싣고 내릴 수 있는 시설, 철광석이나 석탄을 싣고 내리기도 하고, 또 어떤 곳에는 유조선이나 가스를 싣고 내리는 시설이 있다. 물론 콘테이너 부두도 있다.

중요한 사실은 이러한 부두 시설은 제각각 지정된 배만이 접안을 하게 된다. 콘테이너 부두에 여객선이 들어가지 않고, 곡물을 하역하는 부두에 콘테이너 선이 들어가서는 안 된다. 그러므로, 대부분의 항구에서는 각 부두에 번호를 정하고, 번호 범위에 따라 어떤 배들이 접안할 수 있는지 안내를 한다.

인터넷 서버에도 이와 비슷한 개념이 존재한다. 바로 port라는 개념이다. 80번 port는 인터넷 홈페이지 서비스인 http가 사용하는 port이다. 21번 포트는 ftp가 사용하는 포트이며, 전자우편을 보내는 SMTP는 25번 포트를 사용한다.

즉, 인터넷 정보의 종류가 무엇이냐에 따라 서버에 접속하는 포트가 결정이 되는 것이다. 그리고 이것을 인터넷 상의 국제적인 표준인 RFC로 제정하여 모두가 지키고 있다. 공교롭고 재미있는 사실은 부두를 뜻하는 영어 단어가 port 이다.

배의 종류를 인터넷 서비스와 정보의 종류에 비유한다면, 이 배가 접안할 수 있는 부두는 바로 인터넷 서버의 포트가 되는 것이다. 컨테이너 부두가 없는 항구에 컨테이너 선이 들어갈 수 있는가? 답은 당연히 들어갈 수 없다. 마찬가지로 인터넷 서버에도 80번 포트가 없으면 홈페이지를 보는 http서비스는 받을 수가 없다.

이제 다시 우리의 항구 얘기를 해보자. 외국에서 곡물을 실어가기 위한 배가 들어왔다. 그렇다면, 이 배를 타고 온 사람 중에 혹시 위험한 인물이 있지는 않은지 검사를 할 것이다. 또한 외국에서 이 배에 실었던 곡물과 함께 구제역이나 콜레라 등과 같은 무서운 전염병이 묻어오지 않았나 검사하고, 철저한 소독을 실시하게 할 것이다.

여객선이 들어왔을 때는 이보다 더 철저할 것이다. 각 사람의 신분과 여권, 입국허가증을 미리 받아야 하며, 전염병이 퍼져 있는 나라로부터 온 사람이 있다면, 따로 격리해서 검역을 받고, 가지고 온 짐 중에 밀수품이나 마약과 같은 지녀서는 안 되는 물건이 있는지 철저히 검사를 한다.

인터넷 서버도 이와 유사한 기능을 갖고 있다. 각 포트마다 고유의 데몬[5]이라고 하는 프로그램이 그곳을 지키고 있다. 80번 포트에는 httpd 라는 데몬 프로그램이 이 곳을 지켜서 홈페이지에 접속하는 사용자의 권한을 검사한다. 만약 이 httpd 프로그램이 불법적인 홈페이지 접근을 발견하면 접근한 사람에게 "Forbidden" 이라는 금지의 메시지와 함께 홈페이지의 접근을 차단한다. 이는 입국허가증이 없이 들어온 외국인을 발견해서 국외 추방하는 것이나 마찬가지다.

그런데 만약 이런 곳을 지키는 사람이 뇌물을 받거나, 혹은 문제가 생겨서 제대로 검사를 하지 못하는 현상이 발생할 수도 있을 것이다. 이렇게 되면, 그 항구나 입국장을 통해서 검사가 제대로 이뤄지지 않은 사람과 물건이 오갈 수 있게 되고, 이는 곳 범죄행위의 환경을 제공하게 된다.

인터넷 서버의 데몬 프로그램도 문제가 생기거나, 고의로 다른 사람에 의해 바꿔치기 당한다면 이 인터넷 서버로 들어오는 인터넷 서비스를 제어하지 못하게 되며, 불법적인 크래킹을 적발하지 못하게 된다.

2. 인터넷 서비스의 흐름과 해킹의 유형

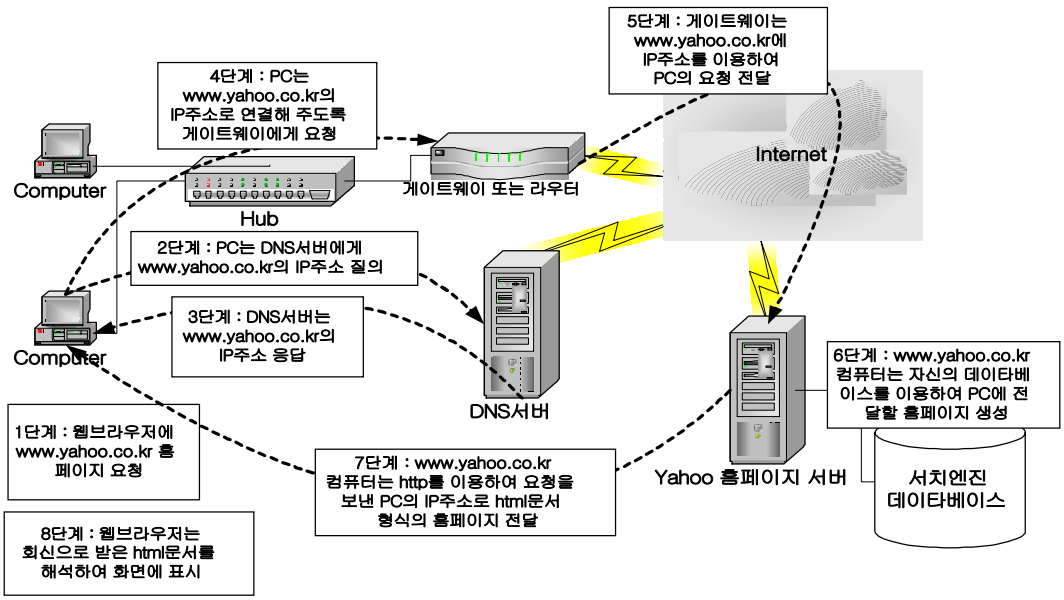

그림 1. 홈페이지 접속으로 본 인터넷 서비스의 흐름도

---

5) **daemon**: 디몬 또는 데몬 이라고 읽으며, 원 영어 사전에서는 demon, 즉 악마와 같은 말로 나온다. 하지만, 또 다른 뜻으로, 수호신, 화신이라는 의미가 있으며, 이 수호신의 의미로 사용할 때는 demon대신에 daemon으로 적게 된다.

앞 그림은 홈페이지의 접속 과정을 통해 살펴본 개략적인 인터넷 서비스의 과정이다. 이 과정에서 설명한 것과 같이 DNS, 웹서버 등의 컴퓨터에는 해당되는 인터넷 서비스를 제공하는 데몬 프로그램과 포트가 열려 있어야 한다. 시스템의 침입자는 이런 과정에서 세 가지 방향의 접근을 한다.

첫 번째는 내 컴퓨터와 서버 사이에 오가는 정보를 훔쳐내는 것이다. 이는 서비스 과정에 거쳐가게 되는 허브, 게이트웨이, 라우터 등의 장비에 접근하여 정보가 오가는 과정을 감시하다가 암호나 보안과 관련된 정보를 주고받는 순간에 그 데이터를 저장하는 수법을 사용한다. 과거 전화를 이용한 모뎀이 통신의 대부분일 때, 스파이들이 많이 사용하는 도청기법이 되겠다. 하지만 이 방법은 암호기술의 발달로 인해 최근에는 거의 사용되지 않는다.

두 번째는 서버 측에 침입자가 내 컴퓨터를 가장하여 접근하는 것이다. 내 컴퓨터에서 텔넷을 접속하여 작업을 하고 있는데, 침입자가 중간에 이 연결을 가로채는 방법을 사용한다. 이렇게 하면 서버 쪽에서는 연결이 계속 이뤄지고 있지만, 내 컴퓨터에서는 서버가 연결을 끊은 것으로 나오고, 실제 이 연결은 중간에 이 연결을 가로챈 침입자가 사용하고 있다.

세 번째는 내 컴퓨터에게 서버를 가장하여, 암호 정보를 가로채는 방법이다. 침입자는 가짜 서버를 만들고, 내 컴퓨터가 가짜 서버에게 자료를 보내도록 하는 방법이다.

그러나 이런 세 가지 방법은 특정한 컴퓨터와 특정한 사용자를 대상으로 하는 스파이 행위로서 첩보영화에서나 나옴직하다. 실제 자신이 중요한 보안 정보를 다루는 사람이 아니라면, 이와 같은 내용의 침입시도는 별로 당할 일이 없을 것이다.

그래도 이와 같은 세 가지 침입시도를 막기 위해서는 인터넷 상에 데이터나 암호를 전송할 때 주의를 요하는 것이다. 이를 위한 프로그램들로는 ssh, ssl 등이 있으며, 자신의 username과 암호를 되도록 공중의 인터넷으로 전송하지 않는 것이 필요하다.

이제 남은 한가지는 프로그램 상의 오류이다. 앞 절에서 설명했듯이 포트를 지키는 daemon 프로그램에 오류가 있다면, 이 때는 앞의 세 가지 예와 관계없이 직접 daemon프로그램의 오류를 이용하여 시스템에 무사통과 할 수 있다. 이런 공격을 포트스캔이라고 한다.

IV. 침입탐지와 보안대책

1. 리눅스 세계의 총기난사 사건, 포트 스캔

최근에 들어 해킹의 양상이 달라졌다. 이전에는 알려진 시스템의 버그나 암호 도청을 통해 해킹을 하는 것이 대부분이었지만, 지금은 이런 일들이 고속화, 자동화되었다. 인터넷에 연결된 컴퓨터에 상대방 시스템의 버그가 있는 포트를 찾아주는 프로그램을 설치하고, 목적지 네트워크 주소만 입력하면, 상대방 컴퓨터에서 문제가 있는 daemon프로그램이 지키고 있는 포트나, 아예 지키고 있지 않는 포트를 찾아내 준다.

최초로 이런 것을 자동화한 유명한 프로그램은 Satan (Security Administrator's Tool for Analyzing Networks) 이다. 이름에서도 알 수 있듯이 이 프로그램은 네트워크 관리자가 자신이 관리하는 네트워크에 혹시 문제점이 있지는 않는가 찾아주는 프로그램으로 만들어 졌다. 이 프로그램은 root로 설치하고, 콘솔의 X윈도우 화면에서 웹브라우저를 이용하여, 수행하도록 만들어져 있다.

지금에 와서는 Satan보다 강력하고 더 많은 (특히 리눅스 상의) 버그들을 찾아 낼 수 있는 비공식적인 포트스캔 프로그램이 소스와 함께 많이 공개되어 있다.[6]

2. 내 컴퓨터에 불법적인 접근을 찾아내자

이제는 시스템의 관리자로서 나의 네트워크로 유입되는 불법적인 접근 시도를 원천 봉쇄할 필요가 있겠다. 그렇다면 가장 먼저 포트스캔 시도를 찾아내는 프로그램을 배워보자. 포트스캔 시도를 찾아주는 프로그램으로 강력하고, 가장 쉬운 인터페이스를 자랑하는 것이 snort 이다. 이 외에도 데비안에서는 portsentry 등의 보다 강력하고, 포트스캔 검사 뿐만 아니라, 접근 차단까지 가능한 프로그램들이 있지만, 그런 내용들은 http://debianusers.org 에 있는 팁란을 참조하도록 하고, 여기서는 기초가 되는 snort의 사용법만을 다룬다.

먼저 snort 프로그램을 dselect를 이용하여 설치한다. 그리면 환경설정 화면이 나오는데, 자신의 네트워크 주소를 적어주기만 하면 된다. 필자의 경우 IP주소가 166.104.226.57 이며, 서브넷 마스크를 255.255.255.0을 사용한다. 서브넷 마스크가 255.255.255.0는 2진수로 표현하면

11111111.11111111.11111111.00000000

이 된다. 이것을 앞에서부터 숫자를 세면 1이 24개가 나온다. 각자 자신의 서브넷 마스크를 2진수로 바꾼다음 앞에서부터 1의 숫자를 세면, 맨 뒤에 적는 숫자가 된다.

다음 그림은 dselect에서 snort패키지를 찾은 화면과, snort설치시 환경설정 화면이다.

---

6) 지금 이 글을 읽고 이런 프로그램을 테스트해보고 싶어진다면 그만 둘 것을 권한다. 대부분의 포트스캔 프로그램들은 시스템 침입을 밥먹듯이 하는 사람들에 의해 만들어졌고, 이런 프로그램의 대다수에는 뻐꾸기 알이 숨어 있다. 즉, 내가 그 프로그램을 실행하는 순간, 내 컴퓨터는 그 포트스캔 프로그램 제작자에게 완전히 노출되게 되는 경우가 많다.

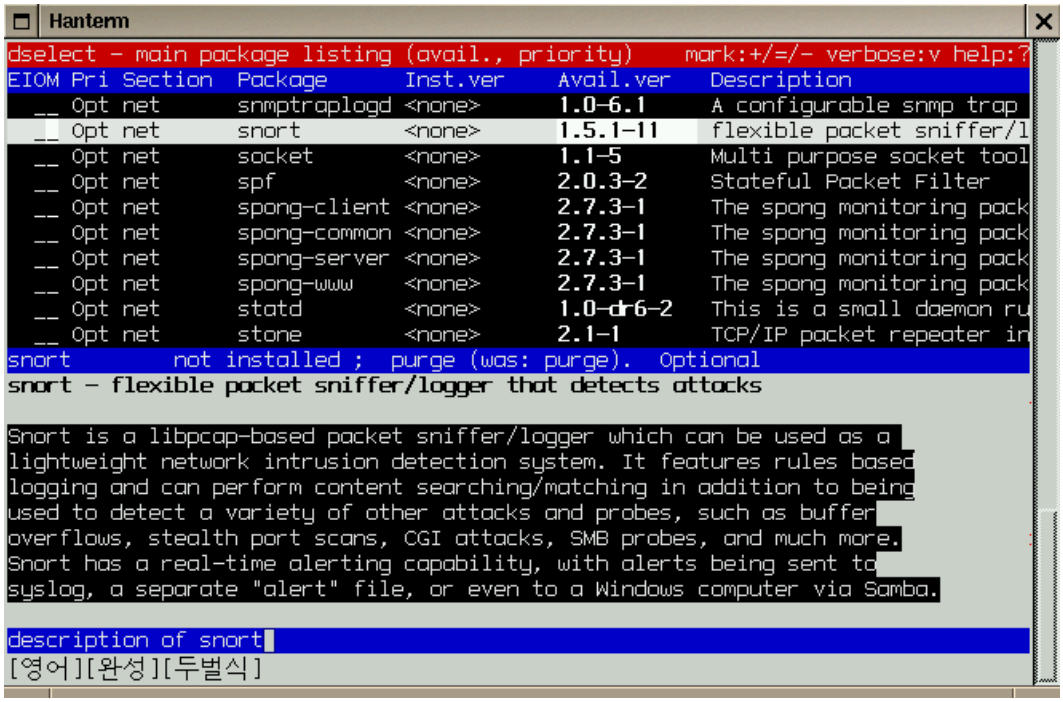

그림 2. 데비안의 dselect프로그램에서 선택 메뉴 중 검색 snort를 찾은 결과

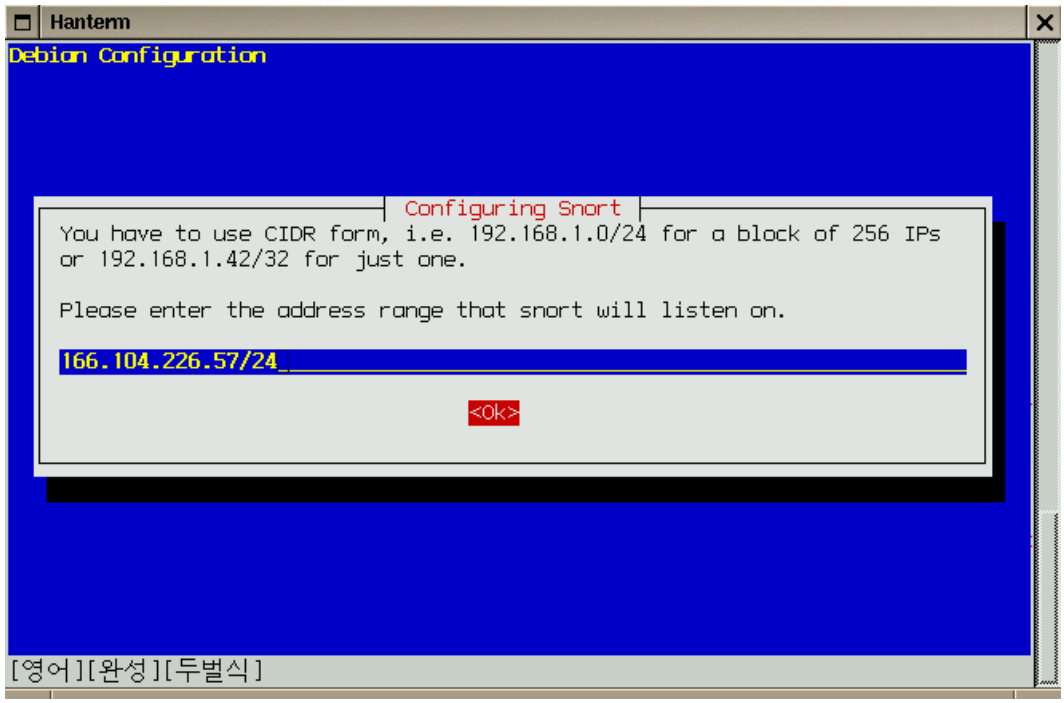

그림 3. snort 설치 시에 물어보는 환경 설정

이렇게 설치해 두면, 이제 하루에 한 번씩 root 에게 다음과 같은 메일이 발송된다.

```
The log begins from: Mar 06 00:01:22
The log ends at:     Mar 06 23:59:25

The number of attack from same host to same destination using same method
==========================================================================
   attacks                    to               from
==========================================================================
1222 Windows Traceroute       166.104.226.255  xxxx39.xxxxxxxx.net
 133 Traceroute               166.104.226.255  xxxxxx.xxxxxxx.ac.kr
 132 Traceroute               166.104.226.255  xxxx.xxxxxxxx.ac.kr
  34 Attempted Sun RPC high   166.104.226.57   xxxxx.xxxxxxxxx.ac.kr
  21 Cgiwrap CGI access atte  166.104.226.68   fuzzy.hanyang.ac.kr
  13 Cgiwrap CGI access atte  166.104.226.57   203.246.15.xxx
  10 Cgiwrap CGI access atte  166.104.226.57   211.200.30.xxx
  10 Cgiwrap CGI access atte  166.104.226.57   211.172.112.xxx
   8 Traceroute               166.104.226.57   xxxxx.xxxx.ru
   7 Attempted Sun RPC high   166.104.226.57   xxxxxx.xxxxx.com
   2 Cgiwrap CGI access atte  166.104.226.57   202.30.50.xxx
```

이 내용 중, 밑줄 친 세 개의 경우는 실제 필자의 서버로 보내진 침입시도이다. Cgiwrap은 CGI의 버그를 이용한 공격이지만, 여기에 나온 것은 필자가 운영하는 CGI게시판에 접근한 정상적인 반복 접근이다. Sun RPC 공격은 리눅스 상에서 NFS 서버를 운영할 때, NFS서버의 버그를 이용하는 공격으로, 최근에 밝혀진 버그임에도 불구하고, 공격 시도가 들어옴을 알 수 있다. Traceroute의 경우에는 직접적인 침입시도라고 보기는 어렵지만, 목적지 주소가 166.104.226.255 인 점을 보아 공격 시도임을 알 수 있다. IP주소가 255인 경우에는 broadcast주소라고 하여, IP 주소가 166.104.226.xxx인 모든 컴퓨터가 이 요청에 반응한다. 이를 통해 현재 해당 서브네트워크 상에 켜져 있는 컴퓨터와 돌아오는 패킷을 분석하여 컴퓨터들의 OS를 알아내려는 시도로 포트스캔의 첫 단계가 되겠다.

이렇게 포트 스캔 시도가 밝혀지면, 개인용 파이어월 프로그램을 이용하여 자신의 시스템에 대한 접근을 차단하거나, 또는 라우터 설정을 변경하여, 해당 침입시도 컴퓨터로부터 내부 네트워크로 유입되는 모든 패킷을 차단하는 등의 방법을 통해 침입자의 접근 시도를 막아야 할 것이다.

3. 성공하는 관리자의 일곱 가지 습관

첫째. 잠깐의 수고로 내일 게으름을 피울 수 있다면, 그 길을 간다.

시스템 관리자는 절대 오늘 게을러서는 안 된다. 내일의 게으름을 위해서라면 오늘 하룻밤쯤은 거뜬하게 새면서, 내일부터 게을러 질 수 있는 도구들을 설치하자. snort같은 경우도 그에 해당된다. 편안히 앉아서 매일 침입시도를 보고 받을 수 있다.

둘째. 두려움을 갖지 않는다.

이런 것은 잘 못하면 시스템을 망가뜨릴지도 모른다고? 이런 두려움은 버리자. 단 한가지 중요한 것은 시스템 자료들을 꼭 백업할 것. 두려움이 없다고 만용하는 것은 망하는 지름길이다. 시스템 자료들을 백업해 두었다면, 과감히 시도해 보다.

누구네 말대로 윈도우즈 98은 아흔 여덟 번 설치해야 안정적이라고 하지 않았나? 망가뜨리고 깔고, 데이터 복구시키고 하는 과정 속에 날로 성장해 가는 자신의 실력을 알 수 있다.

셋째. 찾는 곳에 길이 있다.

찾으라 그러면 열릴 것이다. http://freshmeat.net , http://kldp.org , http://debianusers.org 등등의 홈페이지에서 필요한 자료를 찾으라. 어떤 목적에 사용될 도구를 찾는가? 찾으라 그러면 구할 것이다. 도구를 찾았는데, 사용법을 모르겠나? 찾으라 그러면 찾을 것이다.

넷째. 메일링 리스트를 활용하라.

리눅스와 관련된 메일링 리스트 , 시스템 보안과 관련던 SEC-INFO메일링 리스트 등, 도움을 얻을 수 있는 수많은 메일링 리스트가 있다. 메일이 많이 온다고 귀찮아하지 말고, 이런 메일링 리스트에 가입해 놓자. 적어도 100개 중에 한 개는 내게 소중한 정보가 있다.
(SEC-INFO메일링 리스트 http://cert.cert-cc.or.kr 페이지 참조)

다섯째. 질문을 많이 하라.

잘 모르는 게 있으면 일단 찾는다. 찾아도 모를 때는 지체 없이 질문하라. 메일링리스트도 좋고, 홈페이지 Q&A 도 좋다. 필자처럼 자신이 아는 것을 남김없이 알려주고 싶어 안달난 사람들이 인터넷에는 참 많이도 있다. 단 반드시 찾아보고, 못 찾았을 때나 찾았는데 이해가 가지 않을 때만 질문하라. FAQ도 꼭 먼저 찾아봐라. 이건 네티켓이다.

여섯째. 자기가 하고 있는 일을 기록하라.

두려움이 있는가? 뭔가 새로운 일을 시도하는가? 그렇다면 기록하라. 노트와 파일 두 가지를 병행하면 더욱 좋다. 자신이 입력한 명령어, 설치과정, 설치되는 설정파일들의 위치와 목록 등을 기록해 두라. 사람은 망각의 동물이다.

일곱째. RTFM

많은 사람들이 이 말을 해석하기를 Read The Fucking Manual 이라고 한다. 그러나 나는 이렇게 주장한다. Read The Fine Manual 이라고... 필자는 처음 리눅스를 배울 때, man page를 인쇄하여 공부하였다. 당시에 국내에는 이렇다할 리눅스 책이 전혀 없었고, 리눅스에서 접한 명령어들은 필자가 익히 알고 있던 유닉스 명령어와 옵션이 많이 달랐다.

요즘은 많은 문서들이 있어서 첨부터 매뉴얼만 가지고 배운다는 것은 무식한 일임이 틀림없다. 그러나 너무 많은 사람들이 매뉴얼 읽는 것을 싫어한다. 매뉴얼엔 정답이 있다. 앞에서 찾은 여러 가지 자료를 통해서도 충분한 정보를 얻지 못하였다면 답은 매뉴얼에 있다.

리눅스 상의 많은 매뉴얼들은 man 명령을 통해서 볼 수 있으며, texinfo형식으로 되어 있는 문서는 info 명령을 사용하여 볼 수 있다. 또한 /usr/doc 디렉토리에 가면 각 종 설치된 패키지들의 설치 문서와 설정관련 문서들을 볼 수가 있다.

초고수가 되고 싶은가? 매뉴얼을 읽어라.