iTEASY

# [ Linux Server 보안 설정 5가지 팁]

아이티이지
서비스 운영팀

# - 목 차 -



## ♧ 문서 작성간에 테스트된 환경

- CentOS 5.6 32bit
- Openssh-4.3p2
- iptables-1.3.5
- rkhunter-1.3.8

## ♧ 개요

- 최근 게임 사이트, 포털 사이트 등 개인정보 유출사고가 빈번하게 일어나는 것을 알 수 있습니다. 이와 같은 보안사고는 주요 사이트뿐만 아니라 네트워크가 연결된 모든 서버는 공격 대상이 될 수 있습니다. 서버 보안에 각별한 관심이 필요합니다.

- 해당 문서는 OS 상에서 설정 가능한 보안 설정에 대한 내용을 기술하고 있으며, 서버 관리자 분들께서 서버 운영에 참고해 주시기 바랍니다.

## 1. 서버에서 실행되는 서비스 확인

- 서버에서 실행되는 서비스를 확인하고, 사용하지 않는 서비스는 사용하지 않는 것이 좋습니다. 서버 자원 소모뿐만 아니라, 사용하지 않는 서비스들을 통해 해킹의 위협에 노출될 수 있기 때문입니다.

① 서비스 확인
: 아래 명령어를 통해 현재 서버에 Open 되어 있는 서비스 및 Port 를 확인

root@localhost # netstat –nlp
ex) 기본적인 httpd, mysql, ssh, vsftp 가 실행되어 있는 부분입니다.


※ **서비스별 포트 목록**

| 서비스명 | 기본 포트 | 서비스명 | 기본 포트 |
|---|---|---|---|
| FTP | 21 | IMAP | 143 |
| SSH | 22 | MMS | 554 |
| TELNET | 23 | MSSQL | 1433 |
| SMTP | 25 / 587 | ORACLE | 1521 |
| DNS | 53 | MYSQL | 3306 |
| HTTP | 80 | RDP | 3389 |
| POP3 | 110 | TOMCAT | 8080 |
| HTTPS | 443 | | |

② 부팅시 불필요 서비스 실행 방지

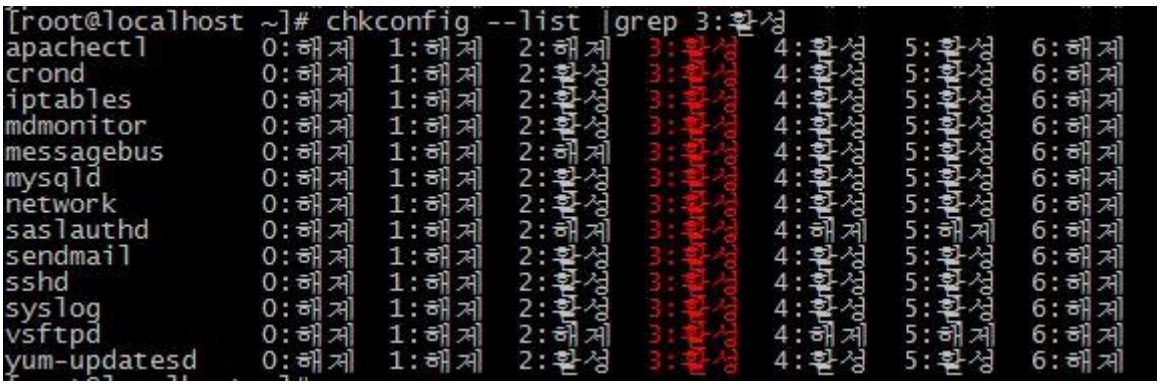
: 사용하지 않는 서비스가 있으실 경우, 서버가 부팅될 때 실행되지 않도록 하는 것이 좋습니다.

root@localhost # chkconfig –list | grep 3:활성

또는

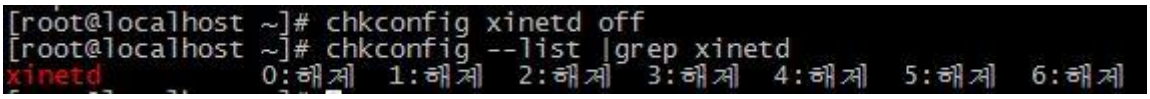
root@localhost # chkconfig –list | grep 3:on

예시) chkconfig –list 중 활성화 되어 있는 서비스들만 확인




root@localhost # chkconfig [서비스명] off

예시) chkconfig 명령어를 이용하여 xinetd 데몬이 재부팅시 자동 실행되지 않도록 설정

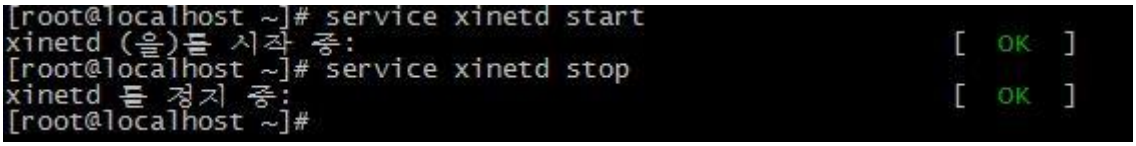



③ 서비스 종료

: 사용하지 않는 프로그램 종료

root@localhost# service [서비스명] stop

예시) service 명령어를 이용하여 xinetd 서비스 종료



## 2. 원격 접속 (SSH(Secure Shell)) 보안 설정

- SSH(Secure shell) 란 telnet 서비스가 보안에 취약한 점을 보완하기 위해서 개발된 것으로, telnet과 달리 주고 받는 패킷들이 모두 암호화 되어 전송되기 때문에 보안에 취약한 점을 보완할 수 있습니다.

- Linux Server에서 사용하는 일반적인 원격 접속 프로토콜 입니다.

※ SSH 설정 파일 경로

- /etc/ssh/sshd_config

※ SSH 서비스 재실행

- root@localhost# services sshd restart

① SSH Root 접근 제한

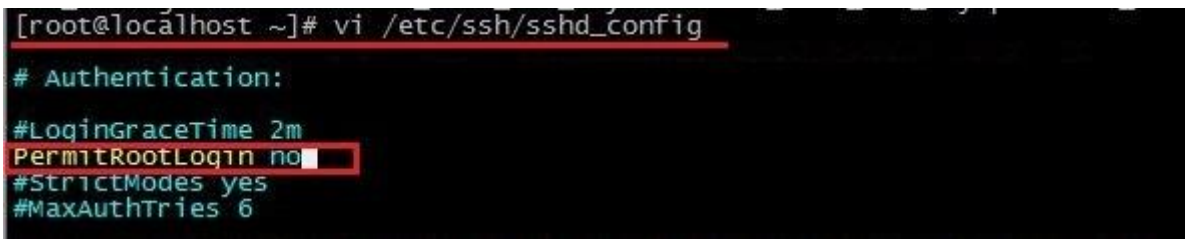
: 초기 ssh 접속시 관리자 계정인 root 접근을 제한

: ssh나 openssh를 사용할 경우 대부분 root로의 직접 로그인을 허용하고 있습니다. 이러한 경우 원격지에서 무차별 대입법 등으로 원격지에서 root 으로 로그인 할 수 있으므로 root로의 직접적인 로그인은 가급적 차단하는 것이 좋습니다.

일반 사용자 계정으로 로그인하여 su 등을 통해 root으로 로그인 하도록 설정하는 것이 좋습니다.

root@localhost# vi /etc/ssh/sshd_config

- PermitRootLogin 를 YES를 NO으로 변경




※ 주의 !! 일반 사용자 계정이 생성되어 있으셔야 합니다.

② SSH 접속 계정 제한

: SSH 으로 접속할 수 있는 사용자 계정을 제한

root@localhost# vi /etc/ssh/sshd_config

➢ AllowUsers [사용자계정명] , 내용 추가

```
[root@localhost home]# vi /etc/ssh/sshd_config

#       $OpenBSD: sshd_config,v 1.73 2005/12/06 22:38:28 reyk Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.
AllowUsers ksidc root
```

root@localhost# useradd ksidc

➢ 일반 사용자 계정 추가

root@localhost# passwd ksidc

➢ 생성한 사용자 계정 패스워드 설정

```
[root@localhost home]# cd
[root@localhost ~]# useradd ksidc
[root@localhost ~]# passwd ksidc
Changing password for user ksidc.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

ksidc@localhost# su –

[root 암호 입력]

➢ SSH 일반 사용자 계정으로 접속하여, root 으로 접속하기

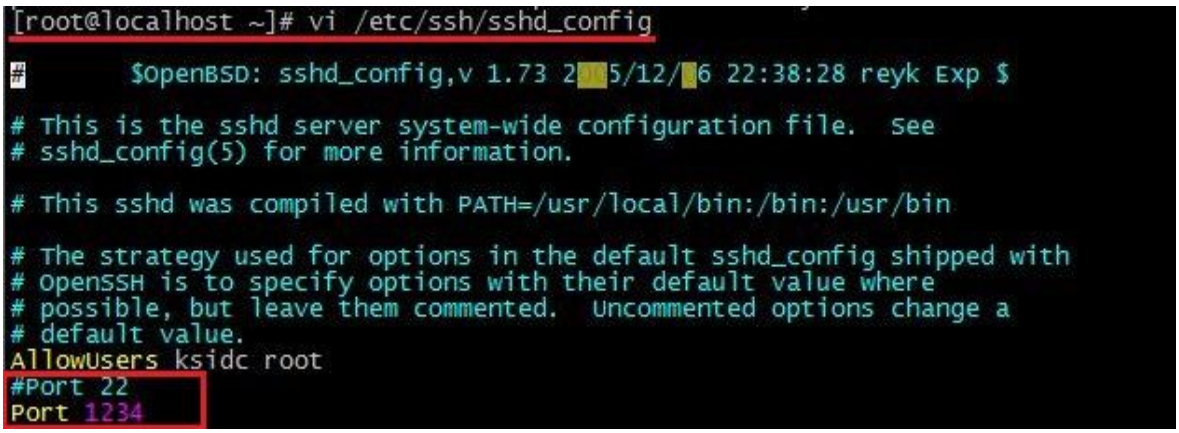
```
[ksidc@localhost ~]$ su -
암호:
[root@localhost ~]#
```

③ SSH 접속 포트 변경

: 기본적인 SSH 포트는 22번 이지만, 반드시 22번 포트를 사용할 필요는 없기 때문에 임의의 번호로 변경하여 외부에서 접속시 변경한 임의 포트로 접속할 수 있도록 합니다.

root@localhost# vi /etc/ssh/sshd_config

➢ Port 22 주석 처리 후, Port [임의 포트] 내용 추가




**※ SSH 설정 파일 변경 후, SSH 서비스를 재실행 하셔야만 설정이 적용됩니다.**

**※ 일반 사용자 계정 ksidc 및 Port 1234는 임의 값으로 절대 똑같이 설정하지 마시기 바랍니다.**

### 3. /tmp 디렉터리 보안 설정

- 웹서버 운영시 /tmp 디렉터리가 필요하며, /tmp 디렉터리는 기본적으로 아무나 읽고, 쓰고, 실행하도록 권한이 설정되어 있습니다. 때문에 웹 서비스를 통해 /tmp 디렉터리에 악성 스크립트를 넣어 실행시킬 수 있으며, 서버 보안에 치명적일 수 있습니다.

① fstab 파일 수정하기

: fstab이란, 리눅스 부팅시 각 파티션으로 마운트하는 정보 및 권한 등에 대한 설정 정보가 있는 파일입니다.

root@localhost# vi /etc/fstab

➢ tmp 설정 중 default로 되어 있는 부분 외에 noexec,nodev,nosuid 를 추가합니다.

```
[root@localhost ~]# vi /etc/fstab

LABEL=/                 /                       ext3    defaults        1 1
LABEL=/backup           /backup                 ext3    defaults        1 2
LABEL=/home             /home                   ext3    defaults        1 2
LABEL=/tmp              /tmp                    ext3    defaults,noexec,nodev,nosuid    1 2
LABEL=/usr              /usr                    ext3    defaults        1 2
LABEL=/var              /var                    ext3    defaults        1 2
LABEL=/boot1            /boot                   ext3    defaults        1 2
tmpfs                   /dev/shm                tmpfs   defaults,noexec         0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
LABEL=SWAP-sda6         swap                    swap    defaults        0 0
```

② /var/tmp 소프트링크 설정

: /tmp 디렉터리 외에도 /var/tmp 디렉터리가 존재하며, 서로 동일하도록 소프트 링크를 설정해 줍니다.

root@localhost# rm –rf /var/tmp

➢ 우선 /var/tmp 디렉터리를 삭제합니다.

root@localhost# ln –s /tmp /var/tmp

➢ /var/tmp 접속시 /tmp 으로 연결되도록 링크를 설정합니다.

**※ /tmp 보안 설정의 경우, /tmp가 별도 파티션으로 구분되어 있어야 합니다.**

**※ mysql의 "mysql.sock"파일 생성 경로가 /tmp으로 설정되어 있을 경우, 홈페이지에서 데이터베이스 접속 장애가 발생할 수 있습니다. 때문에 mysql 설정을 변경하신 뒤 적용하셔야 합니다.**

**※ fstab 설정을 변경한 뒤 서버를 재부팅해야 해당 설정이 적용됩니다.**

## 4. iptable 설정 / hosts.allow & hosts.deny 설정

### I. iptable 설정

- iptable이란 커널에 존재하는 netfilter의 룰을 이용하기 위한 일계의 유틸리티 툴입니다. Netfilter 란 커널에서 패킷을 정책에 따라 필터링 해주는 모듈입니다.

※ iptables 설정 파일 경로

- /etc/sysconfig/iptables

※ iptables 서비스 재실행

- root@localhost# /etc/init.d/iptables restart

① 현재 iptable 설정 확인

root@localhost# iptables –nL

: 서버 방화벽이 설정되어 있지 않을 경우,

```
[root@localhost ~]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

: 서버 방화벽이 설정되어 있는 경우,

```
[root@localhost ~]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
```

② 방화벽 설정

root@localhost# vi /etc/sysconfig/iptables

: vi 편집기를 이용하여 사용하는 서비스 Port 만 등록

```
[root@localhost sysconfig]# vi /etc/sysconfig/iptables

# Manual customization of this file is not recommended.
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
:syn-flood1 - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type any -j DROP
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport 5353 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 21 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 25 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 110 -j ACCEPT
```

**< iptable 옵션 >**

| 옵 션 | 설 명 | 옵 션 | 설 명 |
|---|---|---|---|
| -A | 체인 추가 설정 (INPUT/OUTPUT ) | --sport | 출발지 포트 지정 |
| -s | 출발지 주소 | --dport | 도착지 포트 지정 |
| -d | 목적지 주소 | -j | 규칙 설정<br>(ACCEPT : 허용, DROP : 차단) |
| -m / -p | 사용할 프로토콜 지정<br>(TCP/UDP/ICMP 등) | | |

**Tip) 명령어를 통한 iptables 설정**

: 명령어를 통해 iptables 를 설정할 경우 iptables 서비스를 재실행하지 않으셔도 됩니다.

예시) iptables 에서 웹 서비스 Port 80 을 허용할 경우

root@localhost# iptables -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT

root@localhost# /etc/init.d/iptables save

**※ iptable 설정 파일 변경 후, iptables 서비스를 재실행 하셔야만 설정이 적용됩니다.**

## II. hosts.allow & hosts.deny 설정

- iptable 과 같이 방화벽 기능을 하며, tcp_wrapper를 이용하여 패킷을 필터링합니다.

- 서비스를 재실행 하지 않고, 설정 파일을 수정하면 적용됩니다.

- iptable 과는 달리 포트가 아닌 서비스 명으로 등록 가능합니다.

※ hosts.deny 및 hosts.allow 설정 파일 경로

- /etc/hosts.allow
- /etc/hosts.deny

root@localhost# vi /etc/hosts.allow

: 접속 허용 설정

```
[root@localhost ~]# vi /etc/hosts.allow

#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
sshd : 111.111.111.111
httpd : ALL
vsftpd : 111.111.111.111
```

root@localhost# vi /etc/hosts.deny

: 접속 차단 설정

```
[root@localhost ~]# vi /etc/hosts.deny

#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
sshd : ALL
vsftpd : ALL
```

## 5. 시스템 파일 변조 체크

- 크래커가 서버를 해킹했을 경우, 거의 대부분 이미 백도어 및 루트킷 설치, 시스템 파일을 변조했을 가능성이 높습니다.

- 이러한 시스템에 대한 전체적인 점검이 가능한 프로그램이 있으며, chkrootkit, rkhunter 등이 이에 속합니다.

- chkrootkit 등과 같은 많은 프로그램이 존재하지만 해당 문서에서는 rkhunter에 대해 안내하고 있습니다.

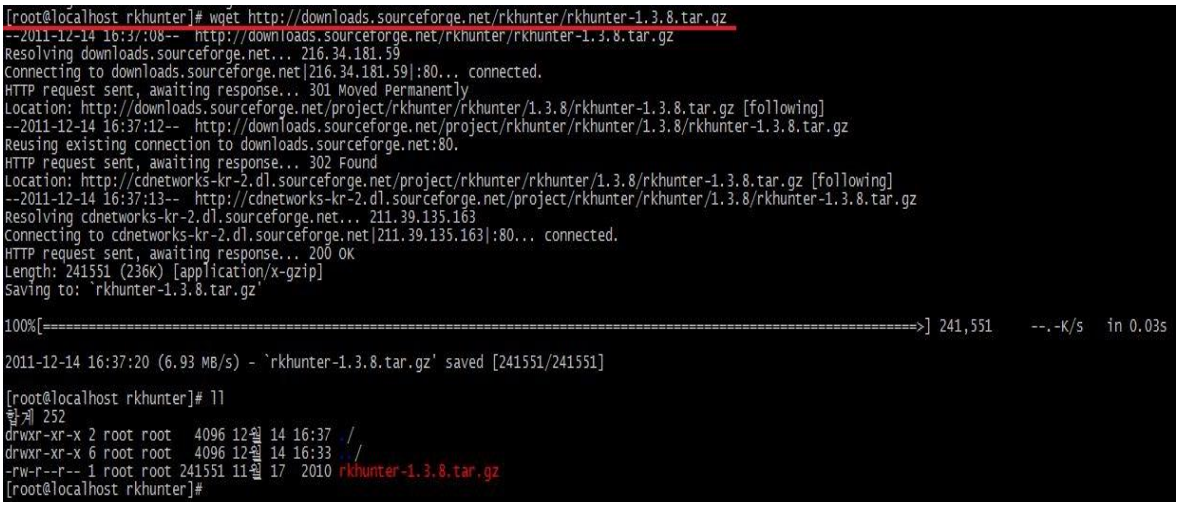
※ 백도어란?
: 공격자가 시스템에 침입한 뒤 차후에 다시 root 등의 최상위 권한으로 접근을 용이하게 하기 위해 사용되는 프로그램이나 도구를 가리킵니다.

※ 루트킷이란?
: 백도어와 같은 프로그램이나 도구의 모음을 가리킵니다.

① rkhunter 다운로드 및 설치

root@localhost# wget http://downloads.sourceforge.net/rkhunter/rkhunter-1.3.8.tar.gz

➢ wget 명령어를 이용하여, 설치 파일을 다운로드 받습니다.



root@localhost# tar xvfz rkhunter-1.3.8.tar.gz

➢ 압축을 해제합니다.

root@localhost # ./installer.sh --layout /usr/local –install

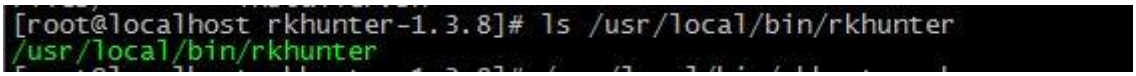
➢ 설치를 진행합니다.

```
[root@localhost rkhunter]# tar xvfz rkhunter-1.3.8.tar.gz
rkhunter-1.3.8/
rkhunter-1.3.8/installer.sh
rkhunter-1.3.8/files/
rkhunter-1.3.8/files/rkhunter.conf
rkhunter-1.3.8/files/LICENSE
rkhunter-1.3.8/files/i18n/
rkhunter-1.3.8/files/i18n/zh.utf8
rkhunter-1.3.8/files/i18n/en
rkhunter-1.3.8/files/i18n/cn
rkhunter-1.3.8/files/i18n/de
rkhunter-1.3.8/files/i18n/zh
rkhunter-1.3.8/files/filehashsha.pl
rkhunter-1.3.8/files/ACKNOWLEDGMENTS
rkhunter-1.3.8/files/contrib/
rkhunter-1.3.8/files/contrib/rkhunter_remote_howto.txt
rkhunter-1.3.8/files/contrib/README.txt
rkhunter-1.3.8/files/contrib/run_rkhunter.sh
rkhunter-1.3.8/files/README
rkhunter-1.3.8/files/CHANGELOG
rkhunter-1.3.8/files/development/
rkhunter-1.3.8/files/development/new-OS-support
rkhunter-1.3.8/files/development/i18nchk
rkhunter-1.3.8/files/mirrors.dat
rkhunter-1.3.8/files/readlink.sh
rkhunter-1.3.8/files/rkhunter.spec
rkhunter-1.3.8/files/stat.pl
rkhunter-1.3.8/files/programs_bad.dat
rkhunter-1.3.8/files/check_modules.pl
rkhunter-1.3.8/files/suspscan.dat
rkhunter-1.3.8/files/rkhunter.8
rkhunter-1.3.8/files/backdoorports.dat
rkhunter-1.3.8/files/rkhunter
rkhunter-1.3.8/files/FAQ
[root@localhost rkhunter]# cd rkhunter-1.3.8
[root@localhost rkhunter-1.3.8]# ./installer.sh --layout /usr/local --install
Checking system for:
 Rootkit Hunter installer files: found
 A web file download command: wget found
Starting installation:
 Checking installation directory "/usr/local": it exists and is writable.
 Checking installation directories:
  Directory /usr/local/share/doc/rkhunter-1.3.8: creating: OK
  Directory /usr/local/share/man/man8: exists and is writable.
  Directory /usr/local/etc: exists and is writable.
  Directory /usr/local/bin: exists and is writable.
  Directory /usr/local/lib64: exists and is writable.
  Directory /var/lib: exists and is writable.
  Directory /usr/local/lib64/rkhunter/scripts: creating: OK
  Directory /var/lib/rkhunter/db: creating: OK
  Directory /var/lib/rkhunter/tmp: creating: OK
  Directory /var/lib/rkhunter/db/i18n: creating: OK
 Installing check_modules.pl: OK
 Installing filehashsha.pl: OK
 Installing stat.pl: OK
 Installing readlink.sh: OK
 Installing backdoorports.dat: OK
 Installing mirrors.dat: OK
 Installing programs_bad.dat: OK
 Installing suspscan.dat: OK
 Installing rkhunter.8: OK
 Installing ACKNOWLEDGMENTS: OK
 Installing CHANGELOG: OK
 Installing FAQ: OK
 Installing LICENSE: OK
 Installing README: OK
 Installing language support files: OK
```

② rkhunter 명령어 확인 및 옵션

root@localhost# ls /usr/local/bin/rkhunter

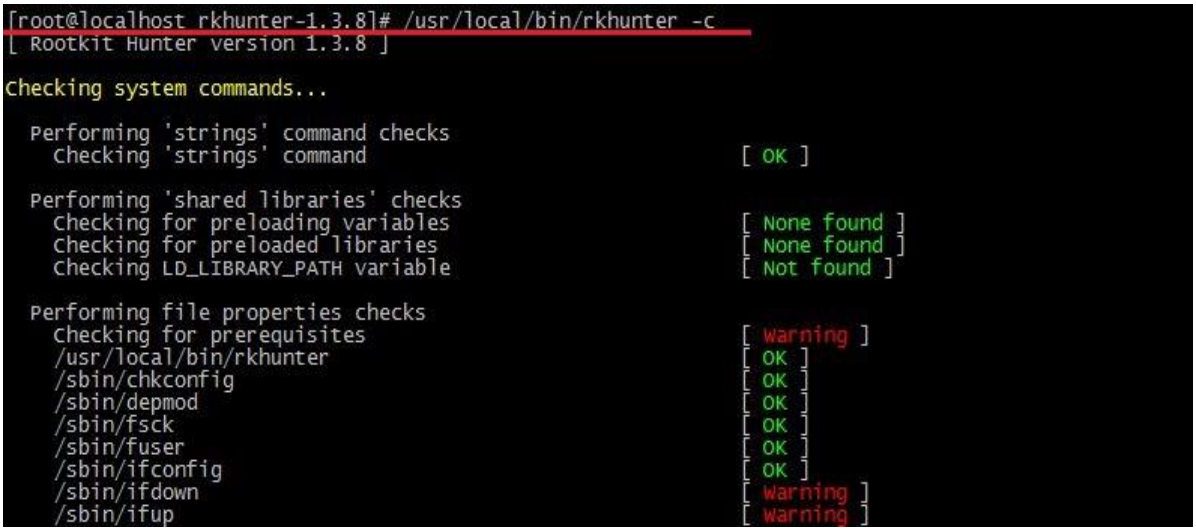
➢ 명령어 위치 확인




| 옵 션 | 설 명 |
|---|---|
| -c | 시스템 체크 |
| --logfile [파일명] | 로그 파일 경로 지정. 기본 /var/log/rkhunter.log |
| -V | rkhunter 버전확인 |
| --update | 업데이트 |

예시) rkhunter 를 이용하여 시스템을 체크

root@localhost# /usr/local/bin/rkhunter -c




**※ rkhunter의 경우 시스템 점검만 가능하며, 변조된 내용의 경우 수동으로 변경하셔야 합니다.**