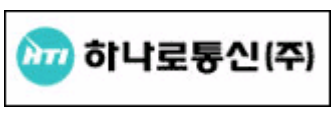
Project: DMZS-Biatchux(F.I.R.E.)

Bootable CD fire-0.4a

2003.5.20

심정재

jjshim AT Hanaro.com



Biatchux(F.I.R.E.) 개요

다른 회사의 시스템 관리자가 당신이 관리하고 있는 시스템이 해킹 피해를 입었을 지도 모른다고 당신에게 알려왔다. 이럴 때 당신은 시스템에서 무엇이 발생되고 있는지 즉시 알고자 한다면, 분석을 도와줄 많은 툴들은 미리 준비되어야 할 것이다. Biatchux는 그러한 툴이다. 침해 사고 대응, 포렌식 분석, 모의 해킹 테스트 등을 수행하도록 새롭게 제작된 CD-ROM으로 부팅 가능한 리눅스 배포판이다. 리눅스 배포판이라고 해서 다른 OS에 대한 포렌식 분석, 데이터 복구, 침해 사고 분석 등을 할 수 없는 것은 아니다. 윈도우 계열 OS와 솔라리스 및 리눅스, FreeBSD등의 다양한 운영체제를 점검 하고 분석할 수 있다. 이 문서는 이러한 Biatchux를 분석한 기술 문서이다. 문서의 마지막 부분에는 포렌식 분석을 수행하고 보안 침해 사고를 대응하기 위한 Biatchux와 유사한 Trinux및 PLAC의 유용성에 대해서도 논의할 것이다.

이 문서를 위해 사용된 Biatchux는 아직도 베타 버전이다(fire-0.4a). 베타 버전이기 때문에 몇몇 버그와 분석을 위한 전체적인 결함이 존재 할 수도 있다. 그러나 Biatchux는 윈도우 NT/2000, Linux, Solaris 시스템 위의 초기 침해사고 대응 및 분석을 수행하기에 매우 뛰어난 툴이다. 또한 포렌식 분석을 위한 깊이 있는 분석 툴들을 리눅스와 솔라리스 시스템 윈도우 시스템 위에서 수행할 수 있도록 정적(static) 바이너리 형태로 제공하고 있다.

Biatchux: 침해 사고 대응을 위한 새로운 툴

소개

이 문서는 Biatchux를 설명한 기술 문서이다. 포렌식 분석을 수행하고 보안 침해 사고를 대응하기 위한 Biatchux의 사용방법과 CD-ROM에 포함되어 있는 다양한 툴을 이해하는 것이다. 궁극적으로 실제 침해사고 발생시 대응할 수 있는 능력을 기르는데 목적이 있다.

Biatchux 개요

Biatchux는 새롭게 제작된 CD-ROM으로 부팅 가능한 리눅스 배포판이다. 개발자(Salusky와 Zendzian)에 의하면 즉각적인 포렌식 분석, 침입 대응, 데이터 복구, 바이러스 검사, 취약점 점검을 수행하도록 디자인 되어 있다고 밝히고 있다. 현재 버전은 fire-0.4a 이며, 약 578MB의 크기를 갖는 ISO이미지로 제공하고 있다. 압축된 ISO이미지를 리눅스 시스템 위에서 다운로드 한 후 CD를 굽기 전에 루프백(loopback) 모드로 이미지를 마운트 하면 이미지의 내용을 볼 수 있다:

```
# mkdir FIRE_image
# mkdir /mnt/image
# cd FIRE_image
# wget http://aleron.dl.sourceforge.net/sourceforge/biatchux/fire-0.4a.iso
# mount -o ro,loop fire-0.4a.iso /mnt/image
```

```
[root@F_I_R_E root]# cd /opt/
[root@F_I_R_E opt]# mkdir FIRE_image
[root@F_I_R_E opt]# cd FIRE_image/
[root@F_I_R_E FIRE_image]# wget http://aleron.dl.sourceforge.net/sourceforge/bia
tchux/fire-0.4a.iso
--14:16:00--  http://aleron.dl.sourceforge.net/sourceforge/biatchux/fire-0.4a.is
o
           => `fire-0.4a.iso'
Resolving aleron.dl.sourceforge.net... done.
Connecting to aleron.dl.sourceforge.net[204.157.3.229]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 606,765,056 [text/plain]

 2% [                                    ] 13,213,320   217.20K/s    ETA 44:28
```

이미지에는 아래와 같은 내용을 포함하고 있다.

```
[root@F_I_R_E FIRE_image]# mount -o ro,loop fire-0.4a.iso /mnt/image/
[root@F_I_R_E FIRE_image]# ls -l /mnt/image/
total 17
-r-xr-xr-x    1 root     root           53 Nov 20  2002 autorun.inf
dr-xr-xr-x    3 root     root         2048 May  6 07:14 docs
-r--r--r--    1 root     root            5 May 15 02:16 FIRE-v0.4a.ver
dr-xr-xr-x    2 root     root         2048 May 14 14:22 fs
dr-xr-xr-x    2 root     root         2048 May 14 14:21 pkgs
dr-xr-xr-x    6 root     root         2048 Feb 16 13:35 statbins
dr-xr-xr-x   22 root     root         8192 May 11 11:16 win32
[root@F_I_R_E FIRE_image]#
```

이미지는 부팅 가능한 리눅스 배포판이다. statbins 디렉터리에는 리눅스 2.4.20 x86커널과 솔라리스 2.7 버전에서 사용 가능하도록 static으로 컴파일한 바이너리 파일(www.incident-response.org 제공)과 윈도우 NT/2000 바이너리파일(Cygwin제공)들이 포함되어 있다. 마지막으로, win32디렉터리에는 윈도우 NT/2000에서 실행 가능한 다양한 보안 응용프로그램들이 있다. Biatchux 0.4a는 Red Hat 리눅스 7.3에 gcc 2.96, kernel version 2.4.20 기반이다.

Biatchux 부팅

다운 받은 이미지를 CD-R에 구운 후 분석할 시스템의 CD-ROM에 Biatchux CD를 삽입한다. 그 다음 분석 시스템의 BIOS설정 메뉴에서 CD-ROM이 가장 먼저 부팅되도록 설정한다. 이 모든 작업이 오류 없이 진행되었다면 CD-ROM으로 부팅 시 초록색 바탕화면에 5개의 메뉴가 나타난다:

1. Fire
2. Fire-serial
3. Fire-X(800 X 600)
4. Fire-X(1024 X 768)
5. Memtest

Biatchux-serial 옵션은 시리얼 콘솔, 터미널 서버나 모뎀을 위한 옵션을 제공하고 있다. Memtest는 x86 PC의 메모리를 테스트 하기 위한 진단 도구를 포함하고 있다.

Biatchux는 /lib와 /usr이 CD-ROM이 필요로 하는 디스크 공간을 최소화 하기 위해 cramfs 파일 시스템을 사용하고 있으며, CD-ROM으로 부팅 할 때 ramdisk에 그들을 로딩 시킨다. 필요한 몇몇 라이브러리에 접근 할 때는 CD-ROM위의 .tgz 파일을 RAM위에서 실행 시키게 되므로, CD-ROM이 시스템으로부터 부팅된 이후에는 드라이브에서

제거될 수 없다.

부팅이 완료되면 “Welcom to F.I.R.E.” 화면이 나타난다(그림1). Biatchux 메뉴의 설정 방법과 포렌식 분석을 하기 위한 많은 옵션을 제공하고 있다는 설명과 관련 로그들이 어디에 쌓이는지에 대한 간단한 이해를 포함하고 있다.

그림 1. F.I.R.E. 초기 환영 메시지

일단 그림1의 환영 메시지 화면을 이해한 후 Enter 키를 누르면 4개의 메뉴가 나타날 것이다.

Start-Here	Basic Config, Networking – etc…
Forensics	Forensics
Virus-Scan	Virus scanning for partition contents
PenTesting	Penetration Assessment tools

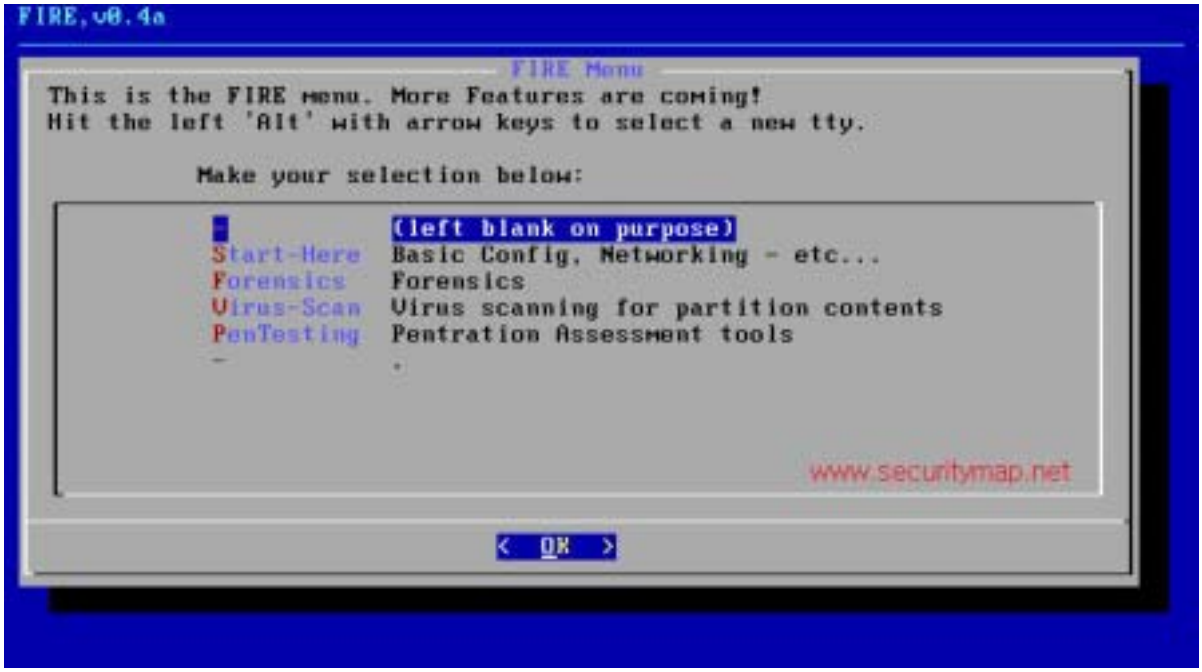


그림 2. Biatchux의 메인 메뉴

대부분의 경우 제일 먼저 Start-Here를 선택한 후, 시작 시 필요한 다양한 네트워크 설정을 하여야 한다(그림 3). 예를 들어 공용 IP를 설정하거나, DHCP를 통해 IP설정을 하여야 한다. 또한 워크스테이션을 원격지에서 접근하기 위한 VNC 서버를 시작 시킬 수 있다. 이것은 운영자가 CD-ROM으로부터 IP설정을 한 이후 VNC 서버를 실행 시키면 조사를 담당하는 사람이 현장에 직접 가지 않고 원격지에서 시스템을 분석할 수 있으므로 매우 유용할 것이다.

그림 3. Start-Here 메뉴화면

MountDataDrv 옵션을 선택하면 드라이브를 마운트 할 수 있는 메뉴(그림 4)로 이동된다. 이 메뉴는 ramdisk에 저장된 데이터를 보다 영구적으로 보관하기 위해 NFS나 SMB를 통해 원격지 저장장치를 마운트 할 수 있는 옵션을 제공하고 있다. 또한 USB-Drive나 Firewire-Drv 마운트 메뉴는 외장 USB 드라이브를 마운트 하거나 USB 저장 장치에 데이터를 비교적 쉽게 옮길 수 있도록 지원하고 있다. 이 메뉴를 통해 마운트를 시도할 때 보여지는 메시지에 따라 커맨드 라인으로부터 이들 장치들을 마운트 할 수 있다.

만약 이들 옵션을 당신이 이용할 수 없다면(예를 들어 외부 저장 장치에 접근하기 쉽지 않은 상태의 경우)배포판에 포함되어 있는 netcat이나 ncftp 혹은 scp를 사용하여 네트워크를 통해 ramdisk의 데이터를 다른 시스템으로 복사할 수 있다. Biatchux 0.4a버전에서는 tfp서버나 Apache 웹 서버를 지원하므로 네트워크를 통해 다른 시스템으로 데이터를 이동하기가 쉬워졌다. netcat을 사용하여 /data 디렉터리를 복사하기 위한 예를 든다면, 아래와 같이 분석 시스템(ex IP: 192.168.10.100) 위에서 리스닝 프로세서를 시작하여야 한다:

```
$ nc –l  -p  3147 > forensic_data.tar
```

Biatchux 커맨드 창에서 data 디렉터리를 보내기 위한 목적지 시스템을 아래와 같이 지정하여야 한다:

```
# cd /data
# tar cf - . | nc  -nv  192.168.10.100 3147
```

중요한 것은 /data 디렉터리를 원격지나 로컬 플로피에 저장하지 않는다면 피해 시스템을 분석하는 동안 RAM에 수집한 모든 정보는 리부팅이나 파워 종료 시 모두 사라질 것이다.

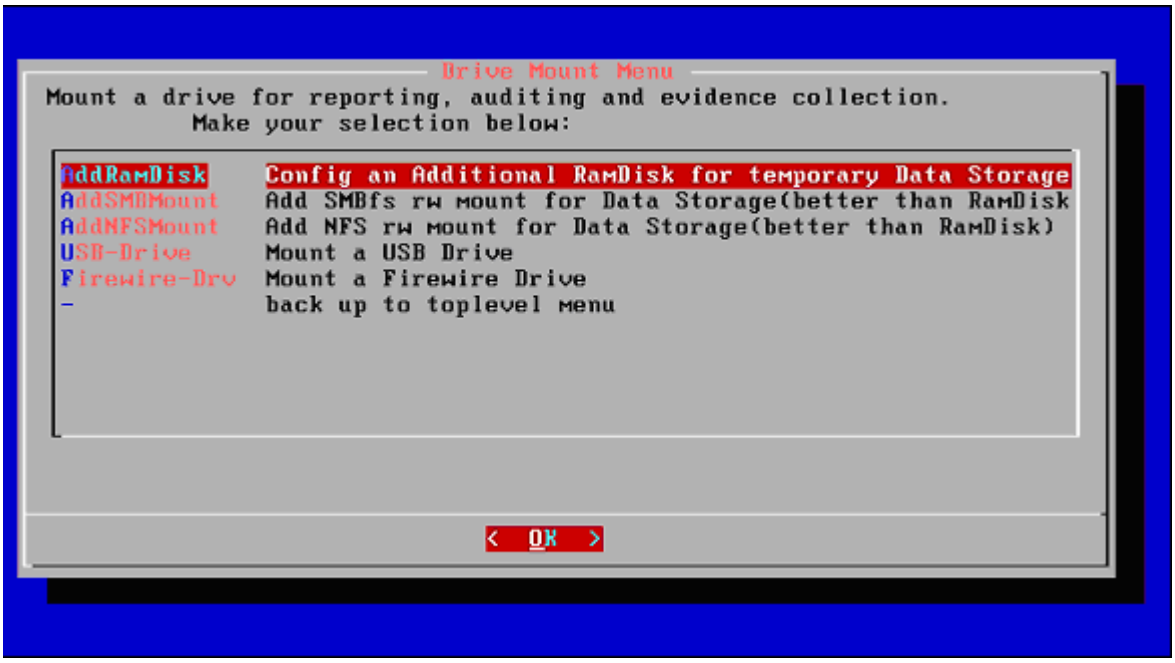


그림 4. Drive Mount 메뉴화면

Start-Here 메뉴로부터 또한 당신은 추가적으로 가상 터미널을 명령 라인으로부터 수행 할 수 있다. 이것은 당신이 중대한 작업을 수행하길 원할 때 주로 사용된다. 각각의 추가적인 가상 터미널은 개별적인 키보드 입력을 수행할 수 있으며 /data 디렉터리에 관련 로그를 기록할 수 있다. 터미널을 바꿀 수 있는 스위치는 ALT키보드와 펑션키(F1~F7)의 조합으로 선택할 수 있다. 첫 번째 가상 터미널(ALT+F1)은 항상 메뉴 방식의 화면이다.

Biatchux 메뉴

현재 모든 로컬 하드 드라이브를 읽기 전용으로 마운트 할 수 있는 능력과 mac-robber 툴을 마운트 된 드라이브에서 수행하고 그 결과 값을 /data에 저장할 수 있는 유용한 포렌식 메뉴를 Biatchux는 지원한다. 또한 Win32 레지스트리와 유닉스 패스워드 파일들을 검색하기 위한 옵션을 가지고 있다. 그 외 루트킷의 설치여부를 확인하거나 이미지 파일에서 검색 수행, 분석되고 있는 드라이브 이미지 백업 본 만들기, autopsy 설정하기 등을 할 수 있다. 현재 모든 옵션이 사용 가능한 것은 아니다.

Virus-Scan 메뉴 옵션은 마운트 된 파일 시스템에 돌아다니는 바이러스를 검색하기 위해 F-Port 유틸리티를 실행한다. F-Port는 DOS, Windows, Linux 바이러스와 트로이잔을 탐지할 수 있는 유틸리티이다. Biatchux는 또한 인터넷으로부터 플로피 디스크로 업데이트된 signature 파일을 로딩할 수 있는 옵션을 제공하고 있다. 업데이트 옵션은 Biatchux가 CD-R미디어로 부팅된 읽기 전용 cramfs 파일 시스템을 사용하고 있어 데이터를 쓸 수 없기 때문에 signature를 업데이트 할 수 없으므로 추가적인 디스크 공간을 마운트 하지 않는 한 아무런 작업도 하지 못할 것이다. 만약 인터넷으로부터 signature를 업데이트 하려고 할 것이라면 디폴트 라우터와 네임 서버를 네트워크 인터페이스에 올바로 설정하여야 한다.

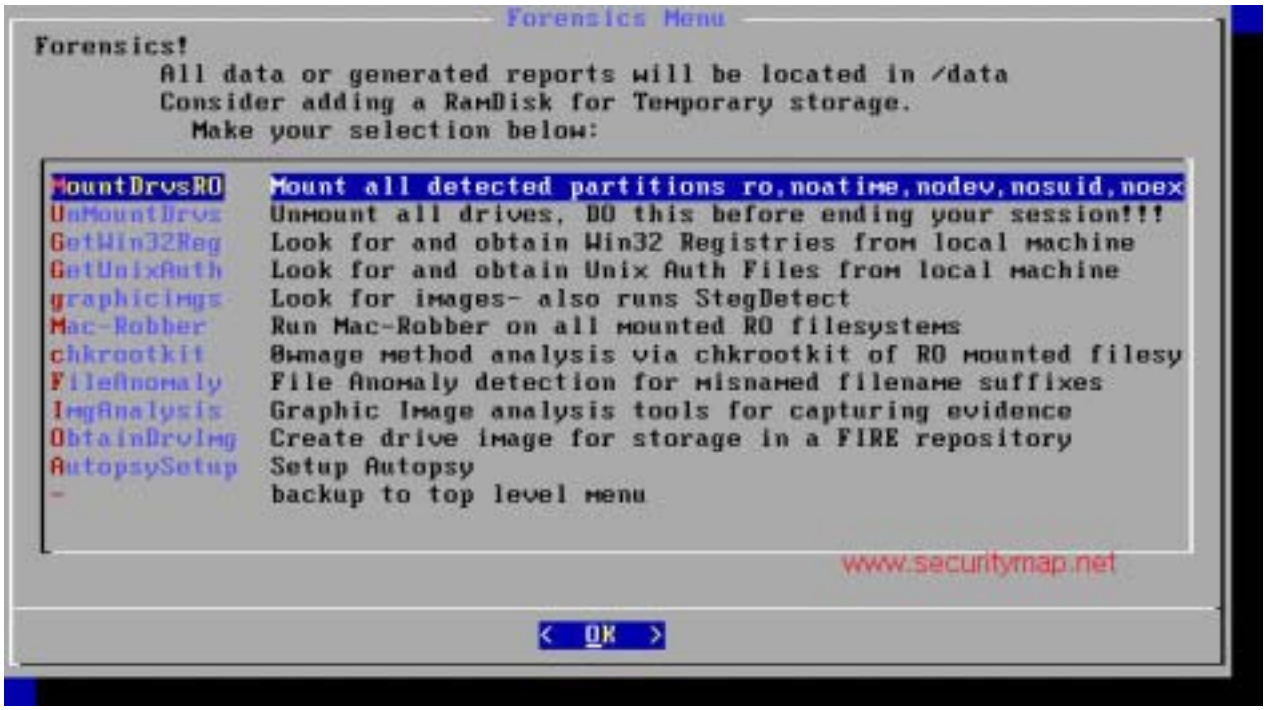


그림 5. Forensics 메뉴화면

그림 6. Virus Scanning 메뉴화면

침입 대응

Biatchux의 빠른 소개를 위해 실제 침입 사고를 대응하는데 어떻게 이용될 수 있는지 볼 것이다. 그 예로서 하나의 시나리오를 소개 하기로 한다. 당신은 의심스런 무언가가 리눅스나 솔라리스 시스템에서 일어난다고 추측하고 시스템을 살펴 보았지만 아무것도 발견 할 수 없었다. 당신은 루트킷이 있을 수 있다고 생각하기 때문에 이 것을

탐지할 믿을 수 있는 몇몇 도구를 필요로 할 것이다. Biatchux는 static 바이너리 파일들을 모아 CD-ROM으로 제공하고 있다. 그래서 당신은 이 신뢰할 만한 Biatchux 도구로 시스템을 살펴 볼 것이다.

당신의 시스템 위에서 어떤 일이 발생되는지 결정하려면 lsof, netstat, ls 그리고 mac-robber(리눅스만 지원)를 사용하길 원할 것이다. Biatchux는 이러한 몇몇 툴을 포함하고 있다. 침입자가 LKM 루트킷을 설치하지 않았다고 가정하면, 이들 도구는 실행중인 프로세서가 무엇인지, 어떤 파일이 사용되고 있는지, 어떤 파일이 새롭게 만들어 졌고, 수정, 접근, 삭제 되었는지, 어떠한 포트가 열려 있는지, 기존에 연결된 네트워크 접속 내역을 볼 수 있다. 또 다른 유용한 도구로 의심스런 바이너리 파일이 변경되었는지 확인하기 위한 md5 체크섬과 원격지 시스템으로 데이터를 복사하기 위한 netcat과 dd유틸리티를 포함하고 있다. md5 유틸리티는 솔라리스 시스템에 특별히 유용하다. 피해 시스템의 md5 체크섬 값을 솔라리스 Fingerprint 데이터 베이스와 비교 검사 할 수 있기 때문이다. Biatchux에 포함된 솔라리스 바이너리는 incident-response.org에서 제공되었으며, lsof, netcat등은 정적으로 연결하지 않았다. 따라서 이 명령의 결과에 대해서는 약간 회의적일 수 있다.

마지막으로 손상된 윈도우 NT/2000 시스템을 조사하는 과정이다. 시스템 전원을 내리기 전에 운영중인 시스템으로부터 어떤 증거를 모을 수 있는지 알고 싶을 것이다. 이러한 시나리오에서 일반적인 OS 배포판에 포함된 침입 대응 툴과 Cygwin 에서 제공하는 바이너리 파일 양쪽 모두 사용할 수 있도록 Biatchux는 CD-ROM으로 제공하고 있다. 조사할 시스템이 CD-ROM 자동실행이 가능하다면 Biatchux CD-ROM을 삽입하면 자동적으로 biatchux.exe가 실행된다. 그렇지 않다면 그림 5의 시작 화면을 보기 위해서는 CD-ROM을 탐색기로 연 후 biatchux.exe를 실행한다. 또한 신뢰된 윈도우 명령 쉘은 cmdenv.bat을 실행시키면 된다.

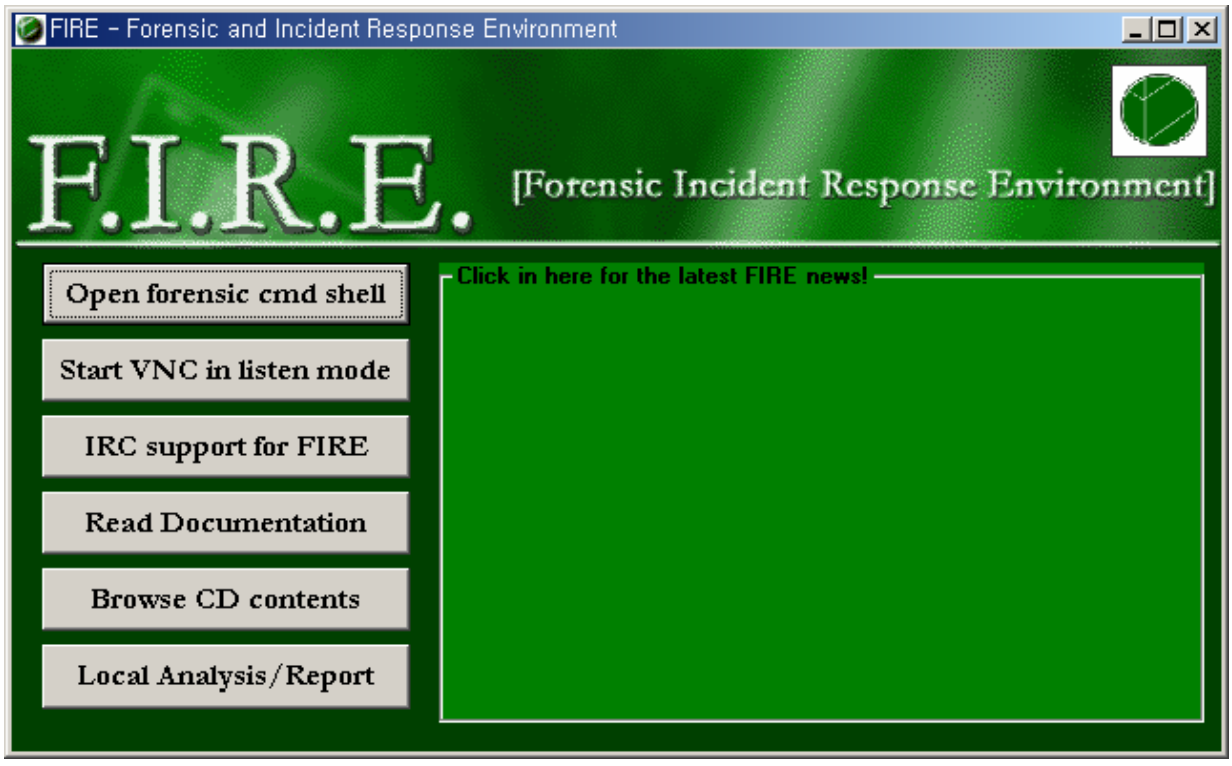


그림 7. F.I.R.E. CD-ROM 자동실행 화면

포렌식 명령 쉘을 배치 파일(cmdenv.bat)로 수행할 수 있다. 배치 파일을 실행하면 CD-ROM의 cmd.exe파일이 수행되며 모든 PATH 변수도 CD-ROM의 디렉터리에서 찾아진다. 그림 8은 배치 파일을 실행한 모습이다.

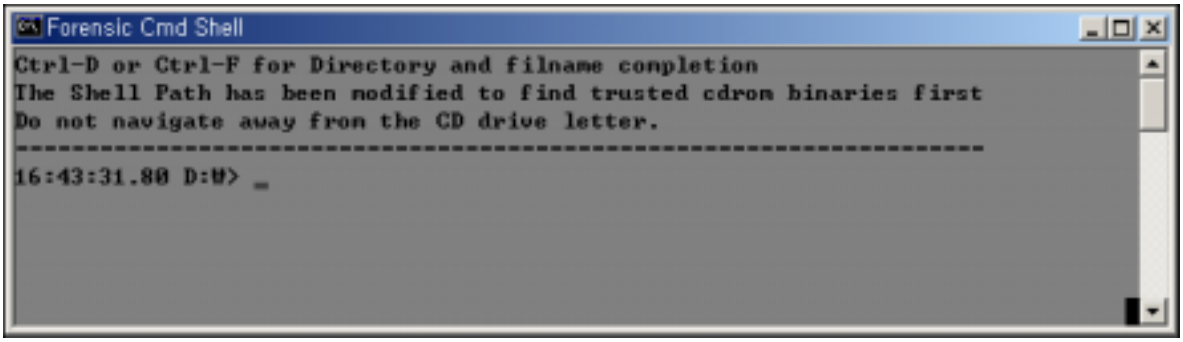


그림 8. Forensics cmd shell

그리고 브라우저를 통해 dmzs.com에 위치한 IRC 서버에 접속할 수 있는 IRC 클라이언트를 제공하고 있다. 또한 리눅스로 부팅할 때 원격 접속을 허용하기 위한 VNC 서버를 시작할 수 있는 것처럼 윈도우에서도 가능하다. 일단 Biatchux의 신뢰된 명령 쉘을 열면 시스템을 분석하기 위한 다양한 도구들을 사용할 수 있다. 아래 그림 9은 Biatchux CD-ROM의 내용이다.

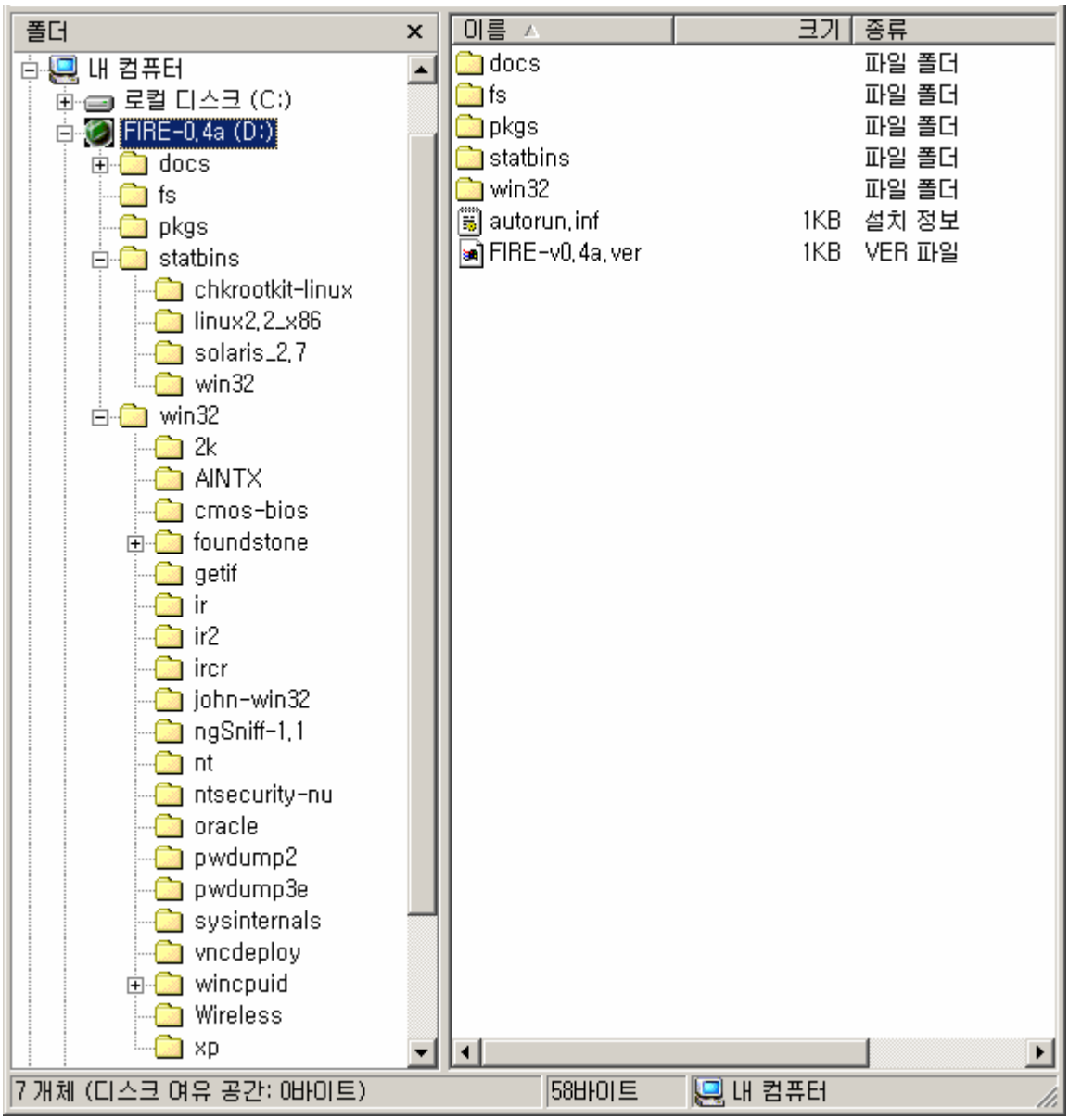


그림 9. F.I.R.E. 의 CD-ROM 디렉터리 구조

● Cygwin 바이너리(Win32 시스템으로 포팅된 유닉스 유틸리티):

ansi2knr.exe	basename.exe	bison.exe	bzip2.exe	bunzip2.exe
bzip2recover.exe	cat.exe	chgrp.exe	chmod.exe	chown.exe
cksum.exe	cmp.exe	comm.exe	compress.exe	cp.exe
csplit.exe	cut.exe	date.exe	dd.exe	df.exe
diff.exe	diff3.exe	dirname.exe	du.exe	echo.exe
egrep.exe	env.exe	expand.exe	expr.exe	factor.exe
fgrep.exe	find.exe	flex.exe	fmt.exe	fold.exe
gawk.exe	make.exe	grep.exe	gsar.exe	gunzip.exe
gzip.exe	head.exe	id.exe	install.exe	join.exe
less.exe	ln.exe	logname.exe	ls.exe	m4.exe
md5sum.exe	mkdir.exe	mkfifo.exe	mknod.exe	mv.exe
mvdir.exe	nl.exe	od.exe	paste.exe	patch.exe
pathchk.exe	pr.exe	printenv.exe	printf.exe	ptx.exe
recode.exe	rm.exe	rman.exe	rmdir.exe	sdiff.exe
sed.exe	seq.exe	sleep.exe	sort.exe	sh.exe
shar.exe	split.exe	stego.exe	su.exe	sum.exe
sync.exe	tac.exe	tail.exe	tar.exe	tee.exe
test.exe	touch.exe	tr.exe	uname.exe	unexpand.exe
uniq.exe	unshar.exe	uudecode.exe	uuencode.exe	wc.exe
wget.exe	which.exe	whoami.exe	xargs.exe	yes.exe
zcat.exe				

● Linux static 컴파일 프로그램(Kernel 2.2 X86)

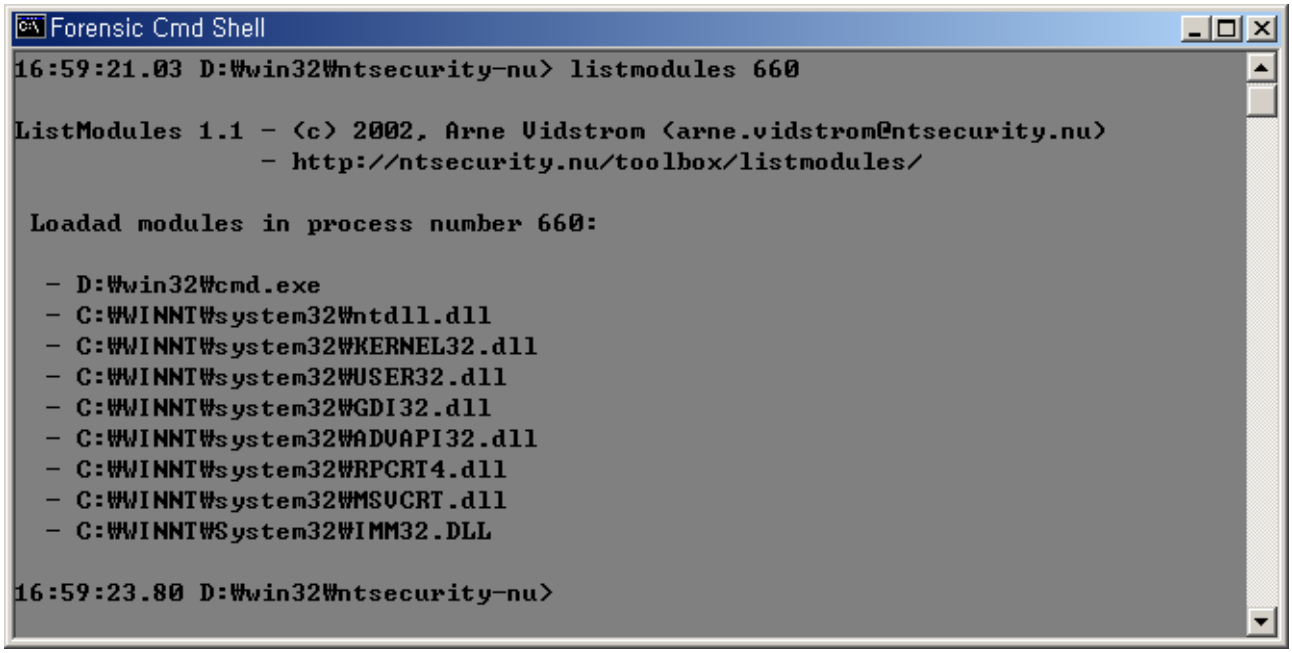
chgrp	chmod	chown	chroot	cksum
cp	cryptcat	cut	date	dd
df	du	echo	env	file
fold	head	hostname	icat	id
ifconfig	ils	ln	ls	lsof
mac-robber	md5	md5sum	mv	nc
netstat	od	pcat	printenv	pwd
rarp	read_data	rm	rmdir	route
search_data	sort	sync	tac	tail
touch	uniq	unrm	uptime	wc
who	whoami	arp	cat	

● Solaris static 컴파일 프로그램(solaris_2.7)

chmod	chown	chroot	cksum	cp
cut	date	dd	df	dirname
du	echo	env	factor	file
gunzip	gzip	head	hostid	hostname
icat	id	ils	join	lastcomm
ln	logname	ls	lsof	md5
md5sum	mkdir	mknod	mv	nc
od	pcat	printenv	pwd	rm
rmdir	rmt	sort	split	su
sum	sync	tail	tar	touch
uname	uniq	unrm	uptime	users
wc	who	whoami	cat	chgrp
zcat				

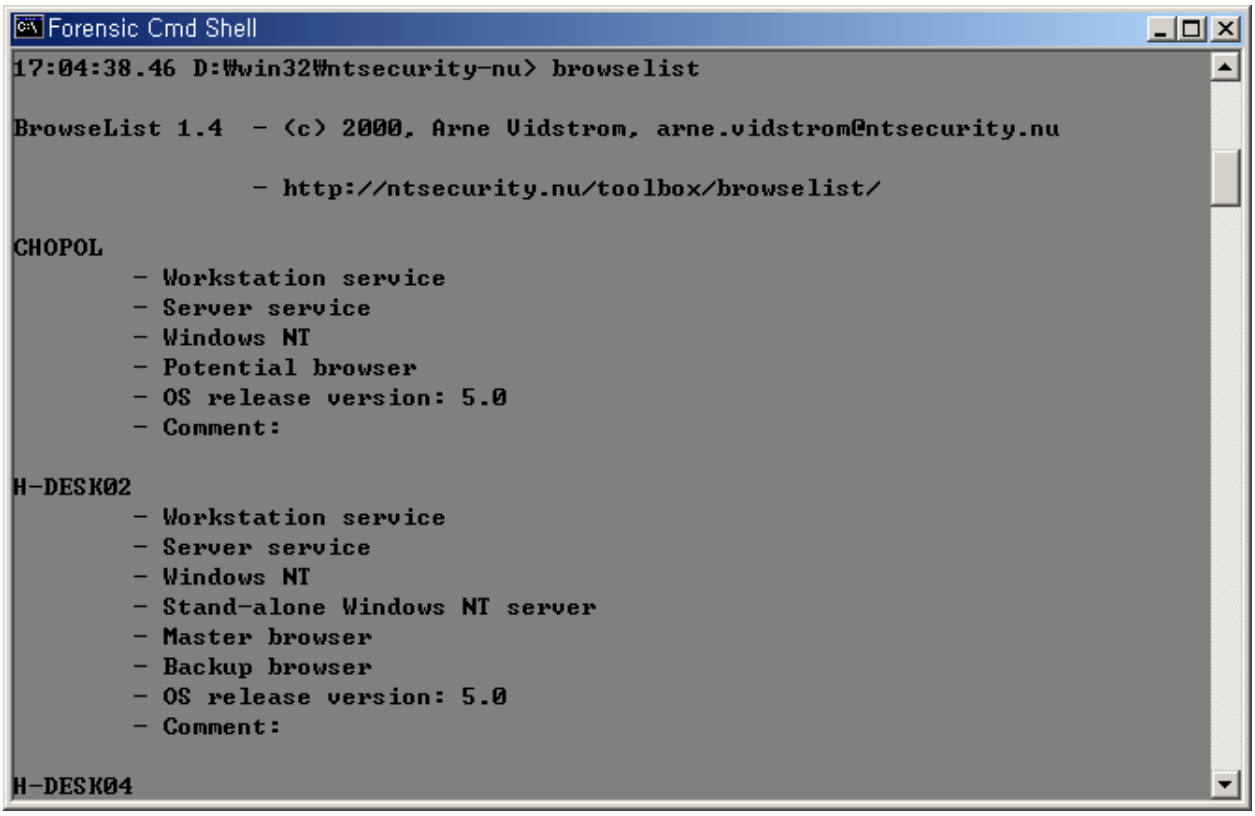
● Win32 유틸리티

디렉터리	유틸리티	설명
ntsecurity-nu	listmodules.exe	Use: 시스템에 로딩된 프로세서 모듈 목록을 보여준다. Syntax: listmodules PID Examples: listmodules 660



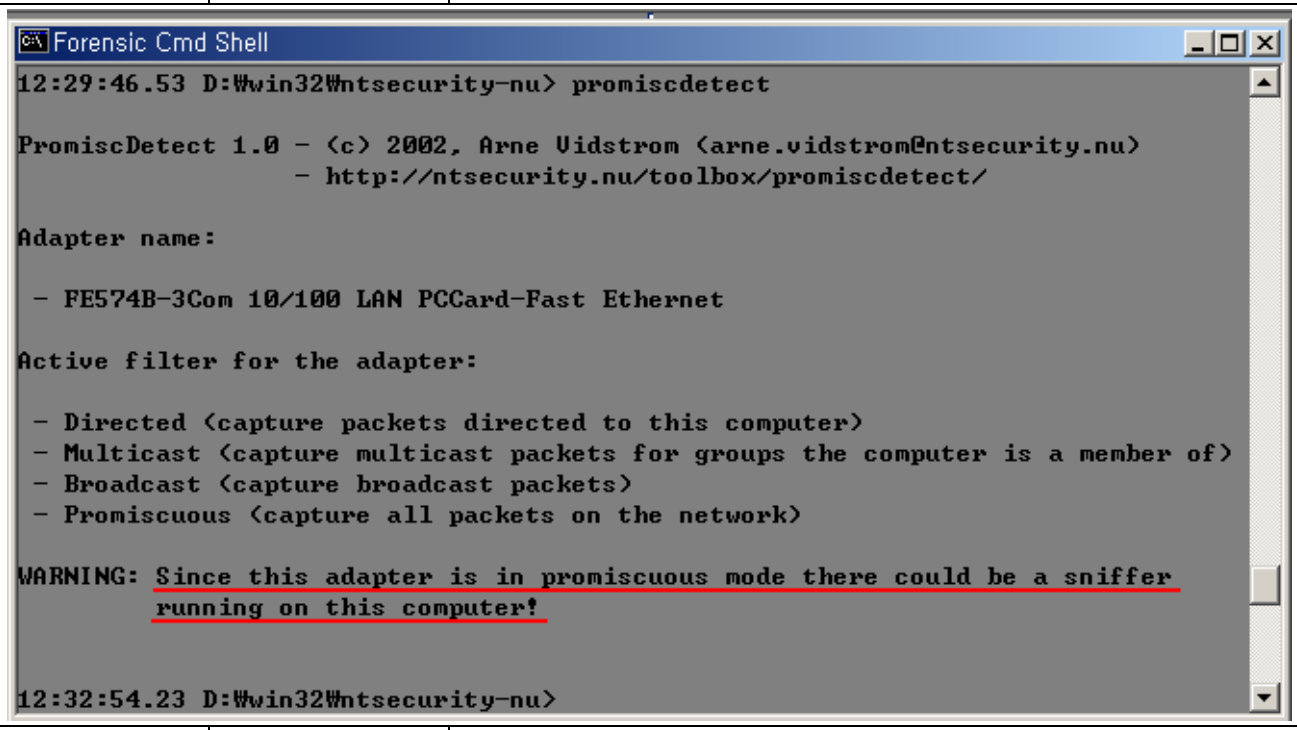
ntsecurity-nu	dumpusers.exe	Use: 로컬 및 원격지 사용자 정보를 덤프 한다. Syntax: DumpUsers -target <computername / IP> -type <notdc/dc> -start <start RID> -stop <stop RID> -mode <verbose/quiet>

ntsecurity-nu	browselist.exe	Use: 동일 Subnet의 윈도우 계열 컴퓨터를 검색한다. Syntax: browselist [Enter] Examples: > browselist [Enter]



ntsecurity-nu	winfo.exe	Use: null session 접속을 통한 윈도우 정보를 수집한다. (계정정보, 트러스트 계정 정보, 공유 폴더 등등) Syntax: winfo IP –n Example: > winfo 192.168.10.100 -n
ntsecurity-nu	gsd.exe	Use: GSD(Get Service Dacl)은 윈도우 서비스에 대한 접근 권한을 확인한다. 즉 각각의 서비스별로 사용자의 엑세스 권한을 확인한다. Syntax: gsd <service name> Examples: > gsd server
ntsecurity-nu	macmatch.exe	Use: 디렉터리 및 파일의 수정, 접근, 생성 시간이 일치하는지 검색한다. Syntax: macmatch <drive/directory> <type> <start date/time> <stop date/time> Examples: macmatch c:\temp -m 2003-04-10:12.01 2003-05-10:12.01
ntsecurity-nu	pmdump.exe	Use: 메모리 내에 로딩된 프로세서 내용을 프로세서 중지 없이 파일로 덤프한다. Syntax: pmdump <pid> <filename> Examples: > pmdump 600 c:\forensics_data\pmdump_600.dmp
ntsecurity-nu	periscope.exe	Use: PE 파일을 정밀하게 검사한다. 파일내부에 악의적인 코드가 실행되는지 확인하며, 파일내의 바이러스 감염 여부 등을 알 수 있다. Syntax: PEriscope <filename> [-q] Examples: > PEriscope Cracking_tools.exe -q

ntsecurity-nu	promiscdetect.exe	Use: NIC의 promiscuous모드 여부를 확인한다. Note: 시스템에서 스니퍼 프로그램이 실행되는지를 확인하는데 사용된다. Syntax: promiscdetect Example: > promiscdetect



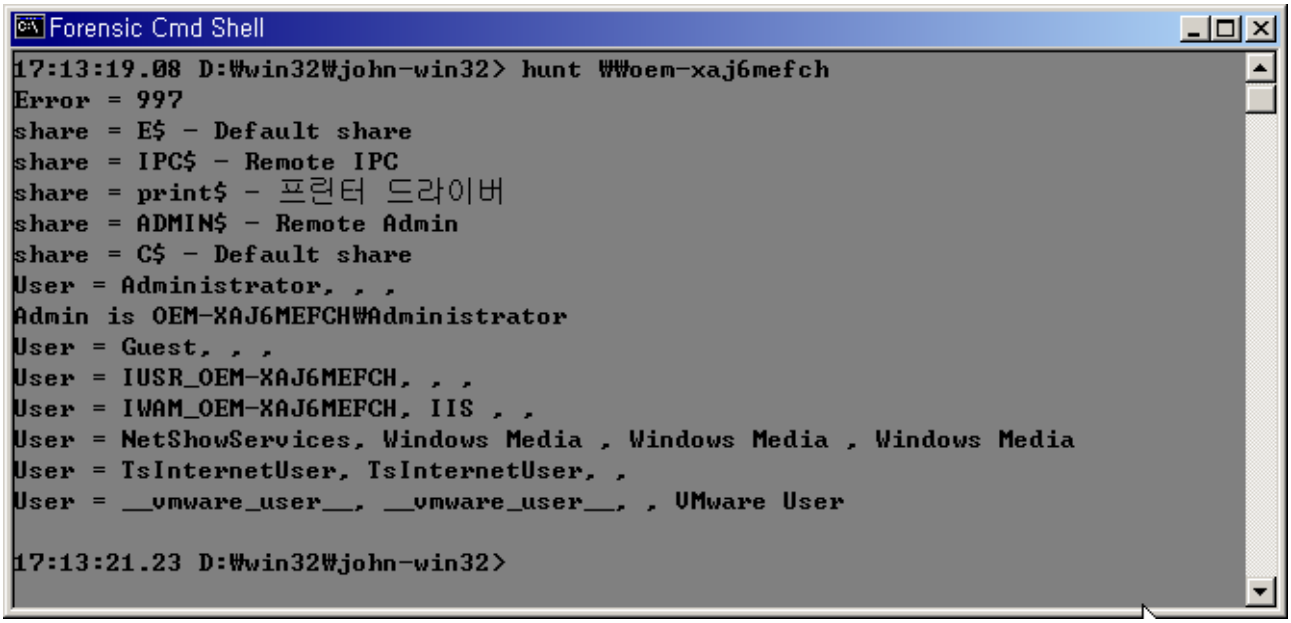
AINTX	abort.exe	Use: 로컬 및 원격지 시스템을 즉시 shutdown 한다. Note: 원격지 시스템을 shutdown하려면 해당 시스템에 대한 관리자 권한을 가지고 있어야 한다. Syntax: abort [-h hostname] Examples: 1) 로컬 시스템 > abort 2) 원격지 시스템 > abort –h \ \ myserver
AINTX	addprpmem	Use: 로컬 및 글로벌 그룹에 맴버를 추가한다. Syntax: addgrpmem [-h hostname] -g group {-u user | -w glbl_grp} Examples: administrators 그룹에 jjshim이라는 사용자를 추가 ➢ addgrpmem –g administrators –u jjshim 로컬 사용자 그룹을 글로벌 그룹 도메인 사용자에 추가 ➢ addprpmem –g users –w “domain users”
AINTX	bootmv.exe	Use: 컴퓨터가 리부팅 될 때 파일 이름을 변경하거나 파일을 삭제한다. Syntax: bootmv -s source_file [-d dest_file] Note: 만약 변경할 파일이름을 지정하지 않으면 소스 파일은 리부팅시 삭제된다. 또한 변경할 목적지 파일은 동일한 디바이스로 지정하여야 하며, 부팅하기 이전에는 변경사항이 적용되지 않는다. Examples: 1) 이름 변경시 ➢ bootmv -s c:\temp\myfile -d c:\temp\rename_file

		2) 파일 삭제시 > bootmv -s c: \ winnt \ system32 \ cracking.dll
AINTX	buevt.exe	Use: 이벤트 로그를 파일로 백업한다. Syntax: buevt [-h hostname] -e event_log -f file_name Examples: 응용프로그램 로그를 백업 > buevt -e Application -f evt_log_bu.evt
AINTX	chsvc.exe	Use: 윈도우 NT4/2000 서비스 설정 사항을 변경한다. Syntax: chsvc [-h hostname] -s service_name options Examples: 1) wins 서비스 "사용안함" 으로 설정 ➢ chsvc –s wins –d 2) wins 서비스 "수동" 시작으로 변경 > chsvc –s wins -m
AINTX	clevt.exe	Use: 이벤트로그를 지운다. Syntax: clevt [-h hostname] -e event_log [-f backup_file] Note: 원격지 로그 파일을 백업하거나 삭제할 수 없다. Examples: 1) 응용프로그램 이벤트 로그 삭제 ➢ clevt -e Application 2) 응용프로그램 이벤트를 저장한 후 삭제 > clevt -e Application -f App_evtlog.evt
AINTX	cpuser.exe	Use: 기존 사용자 등록정보를 기초로 하여 새로운 사용자를 생성한다. Syntax: cpuser [-h hostname] -u username options Examples: administrator 사용자를 한 명 더 만들기 > cpuser –u administrator –n jjshim –f "shim joung jae"
AINTX	df.exe	Use: 디스크의 사용공간과 남은 공간을 보여준다. Syntax: df [drive1 drive2 …] Examples: df c: d:
AINTX	du.exe	Use: 파일 및 디렉터리의 사이즈(KB)와 파일 소유주를 보여준다. Syntax: du [-b -d -o -s -t] file1 ... Examples: du –t –s c: \ temp
AINTX	getprocinfo	Use: 메모리에 로딩된 개별 프로세서 정보를 수집한다. Syntax: getprocinfo [-v] PID Examples: getprocinfo -v 600
AINTX	lsexe.exe	Use: 실행 파일의 헤더 정보를 수집한다. Syntax: lsexe [-s -v] -f file Examples: 1) cracking.dll 파일의 헤더 정보 수집 ➢ lsexe –f cracking.dll ➢ lsexe –f winword.exe | find "Image Version"
AINTX	lssess.exe	Use: 현재 접속된 세션 정보를 상세히 출력한다. Syntax: lssess [-h hostname] [-f] [-c client | -u userid] Examples: lssess -c hacker_pc

AINTX	lsshare.exe	Use: 네트워크를 통해 공유된 자원을 수집한다. Syntax: lsshare [-h hostname] [-acdfp] [-s sharename] Examples: 1) 네트워크 위의 모든 공유된 자원을 수집 ➢ lsshare or lsshare –a 2) 네트워크내의 모든 컴퓨터 이름 출력 > lsshare -c
AINTX	lssvc.exe	Use: 드라이버 등록정보와 서비스 목록을 보여준다. Syntax: lssvc [-h hostname] [-adnsy | -s service_name] Examples: lssvc -yv
AINTX	lstcp.exe	Use: TCP/IP 설정 상태를 보여준다. Syntax: lstcp [-h hostname] Examples: lstcp
AINTX	lsuser.exe	Use: 사용자에 대한 상세정보(사용자 이름, 전체 이름, 마지막 로그인 시간)를 출력한다. Syntax: lsuser [-h hostname] {-u userid | -acnt} Examples: lsuser -a
AINTX	memcheck.exe	Use: 가상 메모리 혹은 물리적 메모리의 전체 사용공간에 대한 메모리 사용내역을 보여준다. Syntax: memcheck Examples: memcheck
AINTX	mksvc.exe	Use: 새로운 윈도우 서비스 항목을 설치한다. Syntax: mksvc [-h hostname] -s service_name options... Examples: remotecmd설치하기 > mksvc -s Remotemcd -p c:\hacking\rcmdsvc.exe -e “Remote Command Service”
AINTX	nbtlookup.exe	Use: Netbios 이름을 IP 주소로 바꾸어 준다. Syntax: nbtlookup computer_name Examples: nbtlookup Hacker_PC
AINTX	portprobe.exe	Use: 원격지 컴퓨터의 TCP/UDP포트에 대한 OPEN 여부를 점검한다. Syntax: portprobe host [-a] begin_port [end_port] Examples: portprobe 192.168.10.100 25-40
AINTX	ps.exe	Use: 현재 실행중인 프로세서에 대한 정보를 출력한다. Syntax: ps [-h hostname] [-ft] [-p procid] Examples: ps –h \ \ Hacking_server or ps –p 23
AINTX	rmshare.exe	Use: 공유된 자원을 제거한다. Syntax: rmshare [-h hostname] -s sharename Examples: rmshare –s foo_drive
AINTX	rmsvc.exe	Use: 윈도우 서비스 제거 Syntax: rmsvc [-h hostname] -s service_name Examples: rmsvc –s Remotecmd

AINTX	rmuser.exe	Use: 사용자 계정 삭제 Syntax: rmuser [-h hostname] -u username Examples: rmuser –h \ \ seriousPC –u cracker
AINTX	shutdown.exe	Use: 로컬 및 원격지 컴퓨터 shutdown Syntax: shutdown [-h hostname] [-fr -m message -w wait] Examples: shutdown -h \ \ myserver -w 60 –m “goodbye”
AINTX	startsvc.exe	Use: 윈도우 서비스 시작하기 Syntax: startsvc [-h hostname] -s service Examples: startsvc –s “IIS Admin Service”
AINTX	stopsvc.exe	Use: 윈도우 서비스 중지하기 Syntax: stopsvc [-h hostname] -s service [-p] Examples: stopsvc –s “IIS Admin Service”
AINTX	strings.exe	Use: 파일내의 문자열을 출력한다 Syntax: strings [-n min_str_len] [-ao] -f file Examples: strings –f hacking.obj
Foundstone	Attacker.exe	시스템의 TCP/UDP 패킷을 통한 접속 현황을 기록한다. 실시간으로 시스템에 접속하는 모든 IP에 대해서 기록 가능하다.
Foundstone	Bintext.exe	파일내의 문자열 중에서 유니코드 및 아스키(Ascii)값을 찾아준다.
Foundstone	boping.exe	Back Oriffice Pinger이다. 시스템내에 알려진 백오리피스를 찾아준다.
Foundstone	ddosping.exe	분산 서비스 거부 툴에 감염되어 있는지 점검하여 준다.
Foundstone	filewatch.exe	특정 파일의 사이즈가 변경 되었는지 모니터링 하여 준다.
Foundstone	fpipe	Use: 소스 포트를 포워딩 하거나 리다이렉트(redirector)하여 준다. TCP/UDP 스트림과 함께 소스 포트를 선택할 수 있다. Syntax: FPipe [-hvu?] [-lrs <port>] [-i IP] IP Examples: fpipe -l 53 -s 53 -r 80 192.168.1.101
Foundstone	fport.exe	Use: 윈도우 시스템에서 열린(opened) TCP/IP 포트에 대해 사용되는 모든 프로세서 목록을 보여준다. Syntax: fport /p /a /I /ap Examples: 1) 포트별로 정렬 ➢ fport /p 2) 응용프로그램으로 정렬 > fport /a
Foundstone	fscan.exe	Use: 명령행 포트 스캐너(sl.exe) Syntax: FScan [-abefhqnv?] [-cditz <n>] [-flo <file>] [-pu <n>[,<n>-<n>]] IP[,IP-IP] Examples: FScan -p 80 192.168.10.3,10.1.2.4,10.1.2.5-10.1.2.20
Foundstone	ntlast.exe	시스템 감사정책이 설정되어 있다면 로그인/로그오프에 대한 성공/실패 여부를 볼 수 있는 뛰어난 툴이다.
Foundstone	afind.exe	특정 시간 동안 접근한 파일들을 결정하기 위해 파일시스템을 검색할 수 있는 유틸리티이다.

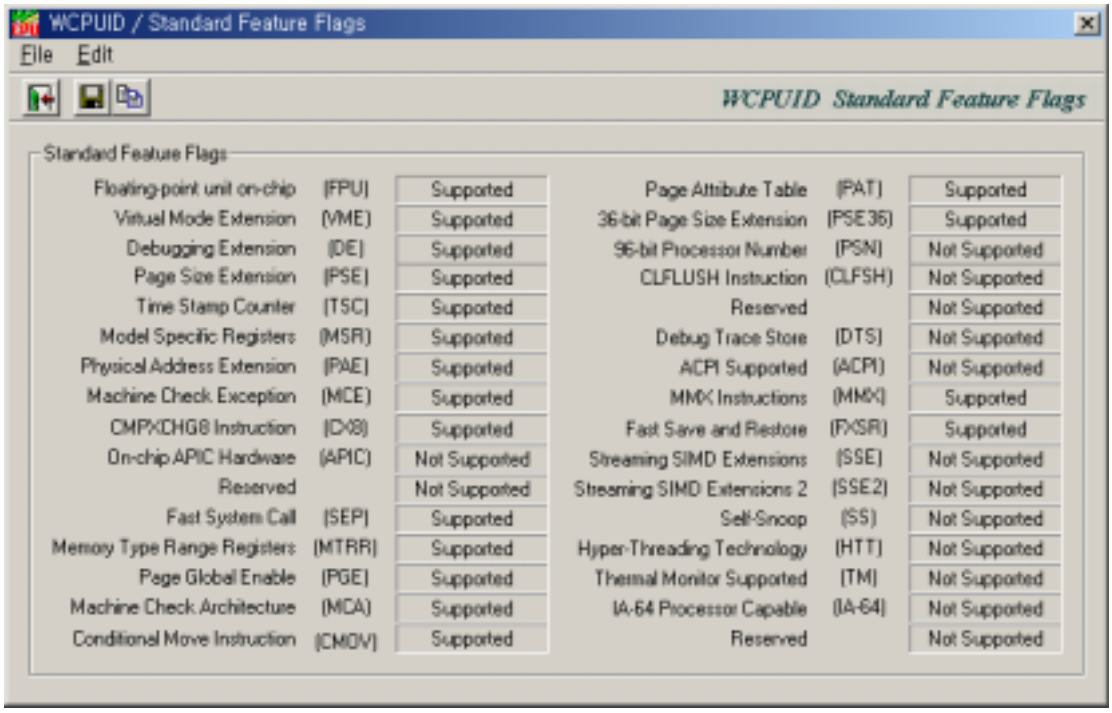
Foundstone	sfind.exe	NTFS 파일 구조에서 숨김 파일들을 찾기 위한 유틸리티
Foundstone	hunt.exe	Use: 원격지 시스템의 공유폴더 이름과 관리자 계정을 찾아준다. Syntax: hunt \ \ servername Exampes: hunt \ \ Serious_PC



getif	getif.2.2.exe	snmp 유틸리티(get, set, getnext, walk, trap 등등)
john-win32	john.exe	Use: 유닉스 패스워드 크래킹 툴(사전 대입 방법) Syntax: john [OPTIONS] [PASSWORD-FILES]
oracle	getsids.exe	Use: 오라클 TNS listener가 요청하는 서비스가 사용하는 오라클 Sid값을 덤프 한다. Syntax: getsids <ip> <port> Examples: getsids 192.168.10.100 1521
pwdump2	pwdump2.exe	Use: 윈도우 NT/2000/XP 의 SAM 데이터 베이스의 패스워드 해쉬를 덤프한다. Note: 로컬 시스템에서 수행하며, 덤프된 SAM 데이터 베이스는 LC3(www.atstake.com/lc3)와 같은 툴로 디코딩 할 수 있다. Syntax: pwdump2 > passdump.txt Examples: pwdump2
pwdump3e	pwdump3e.exe	Use: 윈도우 NT/2000/XP 의 SAM 데이터 베이스의 패스워드 해쉬를 덤프한다. Note: 원격지 시스템의 SAM 데이터 베이스에 해쉬값을 덤프 한다. 해당 시스템의 “Remote Registry Service”가 시작되어야 하며 ADMIN$공유에 접근 가능하여야 한다. Syntax: PWDUMP3E machineName [outputFile] [userName] Examples: pwdump3e \ \ 192.168.10.100
sysinternals	handle.exe	윈도우 NT/2000/XP의 명령 라인에서 handle을 보여준다(Process PID, handle Value, Type, Access Mask, Name)

sysinternals	listdlls.exe	Use: 윈도우 NT/2000/XP의 명령 라인에서 dll 목록을 보여준다(Process PID, Base Address, Mapped Size, Description, Version, Time Stamp, Memory Mapped, Path, Company Name, Image Base Address) Note: 실행된 모든 프로세서와 명령어-라인의 인자 값과 프로세서가 의존하고 있는 라이브러리의 동적 링크된 목록을 보여주는 유틸리티 Syntax: listdlls [-r] [processname|pid] listdlls [-r] [-d dllname] Examples: listdlls –r foo.exe
sysinternals	psexec.exe	Use: Symantec's사의 PcAnywhere처럼 원격지 시스템의 프로그램을 실행시킬 수 있는 원격 제어 프로그램이며 telnet처럼 명령 라인에서 실행한다. Note: 클라이언트 소프트웨어를 원격지 시스템에 설치 하지 않고 단지 시스템에 접근 권한을 가지고 있다면 telnet처럼 사용 가능하다. Syntax: psexec \ \ computer [-u user [-p psswd]][-s][-i][-c [-f]][-d] cmd [arguments] Examples: > psexec \ \ serious_system cmd.exe ➢ psexec \ \ serious_system ipconfig /all ➢ psexec \ \ serious_system –c cracking_tools.exe ➢ psexec \ \ serious_system c:\ bin \ test.exe
sysinternals	psinfo.exe	Use: 로컬 및 원격지 NT/2000/XP의 시스템정보를 수집한다. Note: 커널 빌드 버전, 프로세서 수, Hot-Fix여부 등등 Syntax: psinfo [-h] [-s] [\ \ RemoteComputer] Examples: > psinfo \ \ develop –h
sysinternals	pskill.exe	Use: 로컬 및 원격지 시스템에서 실행중인 프로세서를 중지 시킨다. Syntax: pskill [\ \ RemoteComputer [-u Username]] <process Id or name> Examples: > pskill \ \ serious_com –u guest –p 1234 rcmd
sysinternals	pslist.exe	Use: 목표 시스템 위에서 실행중인 모든 프로세서 목록을 보여준다. Note: 만약 윈도우 NT/2000의 정상적인 프로세서와 악의적인 프로세서 간의 차이점을 말할 수 없다면, pslist 유틸리티는 당신에게 아무 쓸모가 없다. Syntax: pslist [-d][-m][-x][-t][-s [n] [-r n] [\ \ computer [-u username][-p password] [name|pid] Examples: > pslist 503
sysinternals	psloggedon.exe	Use: 원격지와 로컬에 "net"명령을 사용한 접속된 모든 사용자를 보여 준다. Note: 원격지 시스템에 적용하려면 원격 시스템의 "Remote Registry Service"가 시작되어야 하며, 해당 시스템의 관리자 권한이 필요하다. Syntax: psloggedon [-l] [-x] [\ \ computername] Examples: > psloggedon -l
sysinternals	psloglist.exe	Use: 로컬 및 원격지 시스템의 이벤트 로그를 덤프하여 준다. Syntax: psloglist [\ \ RemoteComputer [-u Username [-p Password]]] [-s [-t delimiter]] [-n # | -d #][-c][-x][-r][-a mm/dd/yy][-b

		mm/dd/yy] [-f filter] [-l event log file] <event log> Examples: > psloglist security
sysinternals	psservice.exe	Use: 윈도우 NT/2000/XP의 서비스 환경설정 상태를 수집한다. Syntax: psservice [\ \ Computer [-u Username [-p Password]]] <cmd> <optns> Examples: > psservice \ \ serious –u administrator –p 1234
ngSniff-1.1	ngsniff.exe	Use: 명령 라인에서 스니핑 하기 위한 도구로 패킷 드라이버 설치 없이 W2K이상의 OS에서 통신하는 모든 내용을 덤프하여 준다. Syntax: ngSniff --interface <num> [--help | --list-interfaces | --ignore-addr <host> | -only-addr <host>] Examples: > ngsniff --interface 0 --file c:\ forensics_data \ ngsniff.log
wincpuid	wcpuid.exe	Use: Win32 / Linux / FreeBSD OS에서 시스템에 대한 정보를 하드웨어 레벨에서 살펴 볼 수 있는 유틸리티이다. Syntax: wcpuid.exe Examples: > wcpuid.exe



● 침입자 대응 스크립트(Incident Response Scripts)

: 초기 사고대응을 하는 동안 많은 단계들은 하나의 batch 스크립트로 작성할 수 있다. 사고대응을 위해 자주 사용되는 명령들을 하나의 스크립트를 작성 가능하며, Forensic 시스템에 스크립트 결과를 전송하기 위해서는 netcat을 사용한다. 이것은 윈도우 NT/2000 시스템으로부터 휘발성 데이터의 많은 부분을 저장 시킬 수 있는 간단한 사고 대응 방법이다. 피해 시스템 위에서 메모리를 덤프 하거나 레지스트리로부터 몇몇 정보를 얻어내거나 하는 각각의 서로 다른 일들을 행하고 싶을 것이다. 여기서는 단지 시스템 전원이 꺼지면 잃어 버릴 수 있는 중요한 몇몇 데이터를 획득하는 최소한의 베이스라인(baseline)을 세우는데 목적이 있다.

▶ Windows NT/2000/XP 시스템 침입자 사고 대응 스크립트

: 윈도우 시스템의 네트워크 상태 및 접속 로그, 프로세서 상태를 수집하기 위한 스크립트로 각 OS별 디렉터리에 ir.bat 으로 저장되어 있다.

```
set path=.
cd .. \ windows \ ir
.. \ WinNT \ doskey /history
time /t
date /t
REM request user to input date time
REM datetime
REM network state
.. \ WinNT \ ipconfig /all
promiscdetect
netstat -an
route print
fport
pslist
nbtstat -c
psloggedon
time /t
date /t
.. \ WinNT \ doskey /history
cd .. \ .. \
echo "completed!"
```

: 윈도우 시스템의 이벤트 로그, 사용자 정보, 시스템 정보, 기타 중요 정보를 수집하기 위한 스크립트로 각 OS별 디렉터리에 ir2.bat으로 저장되어 있다.

```
cd .. \ Windows \ ir2
date /t
time /t

REM dump log files
dumpel -l security
dumpel -l application
dumpel -l system
date /t
time /t
REM last sucessful interactive logon
ntlast -l:i -null -n 100
REM last sucessful remote logon
```

```
ntlast -l:r -null -n 100
REM last failed logon
ntlast -f -null -n 100

REM get user info
reg query "HKLM \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ RegisteredOwner"
reg query "HKLM \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ RegisteredOrganization"
reg query "HKLM \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ ProductID"
reg query "HKLM \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ ProfileList"
reg query "HKLM \ SAM \ SAM \ Domains \ Account \ Users \ Names

REM users/groups
net user
net localgroup
REM admins list
REM local administrators



reg query "HKLM \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ Winlogon"

REM get system info
reg query "HKLM \ System \ Controlset001 \ Control \ ComputerName \ ComputerName"
reg query "HKLM \ System \ Controlset001 \ Control \ ActiveComputerName"
reg query "HKLM \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ Currentversion"
reg query "HKLM \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ CSDVersion"
reg query "HKLM \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ WinLogon \ LegalNoticeText"

REM get timezone info

reg query "HKLM \ System \ CurrentControlSet \ Control \ TimeZoneInformation \ StandardName"

REM swap file setting
reg query "HKLM \ System \ CurrentControlSet \ Control \ Session Manager \ Memory
Management \ ClearPageFileAtShutDown"
reg query "HKLM \ System \ CurrentControlSet \ Services \ LanmanServer \ Shares"

REM recent files
reg query "HKLM \ Software \ Microsoft \ Windows \ CurrentVersion \ Explorer \ RecentDocs"

REM startup programs
```

```
reg query "HKLM \ Software \ Microsoft \ Windows \ CurrentVersion \ run"
reg query "HKLM \ Software \ Microsoft \ Windows \ CurrentVersion \ runonce"
reg query "HKLM \ Software \ Microsoft \ Windows \ CurrentVersion \ runonceEx"
reg query "HKLM \ Software \ Microsoft \ Windows \ CurrentVersion \ runServices"
reg query "HKLM \ Software \ Microsoft \ Windows \ CurrentVersion \ runServicesOnce"
reg query "HKLM \ Software \ Microsoft \ Windows NT \ CurrentVersion \ Windows \ Load"
reg query "HKLM \ Software \ Microsoft \ Windows NT \ CurrentVersion \ WinLogon \ Userinit"

REM query Network Interface
REM reg query "HKLM \ System \ CurrentControlSet \ Services \ Tcpip \ Parameters \ Interfaces" /S
reg query "HKLM \ System \ CurrentControlSet \ Services \ Tcpip \ Parameters" /S
REM query Telnet Services
reg query "HKLM \ System \ CurrentControlSet \ Services \ Tlntsvr" /S

REM browser proxy setting
reg query "HKCU \ Software \ Microsoft \ Windows \ CurrentVersion \ Internet Settings"

REM explorer history and settings
reg query "HKCU \ Software \ Microsoft \ Windows \ CurrentVersion \ Explorer \ ComDlg32 \ OpenSaveMRU" /S
reg query "HKCU \ Software \ Microsoft \ Windows \ CurrentVersion \ Explorer \ RunMRU" /S
reg query "HKCU \ Software \ Microsoft \ Windows \ CurrentVersion \ Explorer \ Shell Folders" /S
reg query "HKCU \ Software \ Microsoft \ Windows \ CurrentVersion \ Explorer \ User Shell Folders" /S
cd .. \ .. \
```

▶ 리눅스 시스템 침입자 사고 대응 스크립트

: 리눅스 시스템의 초기 침입자 사고 대응시 휘발성 데이터를 수집하기 위한 스크립트 이다.

```
CURDIR=`./linux2.2_x86/pwd`
BINDIR="$CURDIR/linux2.2_x86"
$BINDIR/echo "All output can be redirected to your choice of storage or netcat pipe."

$BINDIR/echo "========="
$BINDIR/echo "hostname:"
$BINDIR/echo "========="
$BINDIR/hostname
$BINDIR/echo

$BINDIR/echo "========="
$BINDIR/echo "cpu info:"
$BINDIR/echo "========="
$BINDIR/cat /proc/cpuinfo
```

```
$BINDIR/echo

$BINDIR/echo "========="
$BINDIR/echo "disk use:"
$BINDIR/echo "========="
$BINDIR/df -h
$BINDIR/echo

$BINDIR/echo "============="
$BINDIR/echo "fdisk output:"
$BINDIR/echo "============="
fdisk -l
#need a binary for this
$BINDIR/echo

$BINDIR/echo "========="
$BINDIR/echo "version:"
$BINDIR/echo "========="
$BINDIR/cat /proc/version
$BINDIR/echo

$BINDIR/echo "==================="
$BINDIR/echo "kernel boot params:"
$BINDIR/echo "==================="
$BINDIR/cat /proc/cmdline
$BINDIR/echo

$BINDIR/echo "================================="
$BINDIR/echo "Local shell environment variables:"
$BINDIR/echo "================================="
$BINDIR/env
$BINDIR/echo

$BINDIR/echo "========================="
$BINDIR/echo "currently logged in users:"
$BINDIR/echo "========================="
$BINDIR/who
$BINDIR/echo

$BINDIR/echo "========================="
```

```
$BINDIR/echo "List of running processes:"
$BINDIR/echo "========================="
ps -efl
#$BINDIR/ps -efl ; need static ps command

$BINDIR/echo
$BINDIR/echo "======================"
$BINDIR/echo "Network interface info"
$BINDIR/echo "======================"
$BINDIR/ifconfig -a
$BINDIR/echo
$BINDIR/ifconfig -s
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "=================="
$BINDIR/echo "arp table entries:"
$BINDIR/echo "=================="
$BINDIR/arp -n
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "=================="
$BINDIR/echo "/etc/hosts file:"
$BINDIR/echo "=================="
$BINDIR/cat /etc/hosts
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "======================"
$BINDIR/echo "/etc/resolv.conf file:"
$BINDIR/echo "======================"
$BINDIR/cat /etc/resolv.conf
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "======================"
$BINDIR/echo "/etc/passwd file:"
$BINDIR/echo "======================"
$BINDIR/cat /etc/passwd
```

```
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "======================"
$BINDIR/echo "/etc/shadow file:"
$BINDIR/echo "======================"
$BINDIR/cat /etc/shadow
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "==================================="
$BINDIR/echo "netstat output(current connections)"
$BINDIR/echo "==================================="
$BINDIR/netstat -anp
$BINDIR/echo

$BINDIR/echo "=============="
$BINDIR/echo "routing table:"
$BINDIR/echo "=============="
$BINDIR/netstat -rn
$BINDIR/echo

$BINDIR/echo "============================"
$BINDIR/echo "listening ports (via lsof):"
$BINDIR/echo "============================"
$BINDIR/lsof -P -i -n
$BINDIR/echo

$BINDIR/echo "============"
$BINDIR/echo "lsof output:"
$BINDIR/echo "============"
$BINDIR/lsof
$BINDIR/echo

$BINDIR/echo "============"
$BINDIR/echo "Memory info:"
$BINDIR/echo "============"
$BINDIR/cat /proc/meminfo
$BINDIR/echo
$BINDIR/echo
```

```
$BINDIR/echo "============"
$BINDIR/echo "Module info:"
$BINDIR/echo "============"
$BINDIR/cat /proc/modules
$BINDIR/echo
$BINDIR/echo

$BINDIR/echo "==========="
$BINDIR/echo "Mount info:"
$BINDIR/echo "==========="
$BINDIR/cat /proc/mounts
$BINDIR/echo
$BINDIR/echo "Swap info:"
$BINDIR/echo "=========="
$BINDIR/cat /proc/swaps
$BINDIR/echo
$BINDIR/echo "==========="
$BINDIR/echo "    fstab    "
$BINDIR/echo "==========="
$BINDIR/cat /etc/fstab
$BINDIR/echo
$BINDIR/echo



$BINDIR/echo
"================================================================="
$BINDIR/echo "Running Process info:"
$BINDIR/echo
"================================================================="
$BINDIR/ls /proc | sort -n | grep -v [a-z,A-Z] | while read PID
  do
        $BINDIR/echo "Process ID $PID:"
        $BINDIR/echo "/proc/$PID/cmdline:"
        $BINDIR/cat /proc/$PID/cmdline
        $BINDIR/echo
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/environ:"
        $BINDIR/cat /proc/$PID/environ
        $BINDIR/echo
```

```
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/maps:"
        $BINDIR/cat /proc/$PID/maps
        $BINDIR/echo
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/stat:"
        $BINDIR/cat /proc/$PID/stat
        $BINDIR/echo
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/statm:"
        $BINDIR/cat /proc/$PID/statm
        $BINDIR/echo
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/status:"
        $BINDIR/cat /proc/$PID/status
        $BINDIR/echo
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/mem:"
        $BINDIR/cat /proc/$PID/mem
        $BINDIR/echo
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/root:"
        $BINDIR/ls -ld /proc/$PID/root
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/cwd:"
        $BINDIR/ls -ld /proc/$PID/cwd
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/exe:"
        $BINDIR/ls -ld /proc/$PID/exe
        $BINDIR/echo
        $BINDIR/echo "/proc/$PID/fd/*:"
        $BINDIR/ls -lrta /proc/$PID/fd/
        $BINDIR/echo
        $BINDIR/echo "========================================================="
        $BINDIR/echo
  done

$BINDIR/echo "================="
$BINDIR/echo "SUID/SGID search:"
$BINDIR/echo "================="
```

```
$BINDIR/echo
find / -perm -2000 -o -perm -4000 -print | xargs $BINDIR/ls -l {}

$BINDIR/echo "================="
$BINDIR/echo "File permissions:"
$BINDIR/echo "================="
$BINDIR/echo "/etc:"
$BINDIR/echo "-----"
$BINDIR/ls -lrta /etc
$BINDIR/echo
$BINDIR/echo "/bin:"
$BINDIR/echo "-----"
$BINDIR/ls -lrta /bin
$BINDIR/echo
$BINDIR/echo "/sbin:"
$BINDIR/echo "------"
$BINDIR/ls -lrta /sbin
$BINDIR/echo
$BINDIR/echo "/usr:"
$BINDIR/echo "-----"
$BINDIR/ls -Rlrta /usr
$BINDIR/echo
$BINDIR/echo "/var:"
$BINDIR/echo "-----"
$BINDIR/ls -Rlrta /var
$BINDIR/echo
$BINDIR/echo "/dev:"
$BINDIR/echo "-----"
$BINDIR/ls -Rlrta /dev
$BINDIR/echo
$BINDIR/echo "/home:"
$BINDIR/echo "------"
$BINDIR/ls -Rlrta /home
$BINDIR/echo
$BINDIR/echo "/lib:"
$BINDIR/echo "------"
$BINDIR/ls -Rlrta /lib
$BINDIR/echo
$BINDIR/echo "============================================================"
```

```
$BINDIR/echo "================="
$BINDIR/echo "MD5SUMs:"
$BINDIR/echo "================="
$BINDIR/echo "/etc:"
$BINDIR/echo "-----"
$BINDIR/md5sum /etc/*
$BINDIR/echo
$BINDIR/echo "/bin:"
$BINDIR/echo "-----"
$BINDIR/md5sum /bin/*
$BINDIR/echo
$BINDIR/echo "/sbin:"
$BINDIR/echo "------"
$BINDIR/md5sum /sbin/*
$BINDIR/echo
$BINDIR/echo "/usr:"
$BINDIR/echo "-----"
$BINDIR/md5sum /usr/*
$BINDIR/echo
$BINDIR/echo "/var:"
$BINDIR/echo "-----"
$BINDIR/md5sum /var/*
$BINDIR/echo
$BINDIR/echo "============================================================"

#$BINDIR/echo "Make sure to grab /proc/kcore"
```

▶ 솔라리스 시스템 침입자 사고 대응 스크립트

: 솔라리스 시스템의 초기 침입자 사고 대응시 휘발성 데이터를 수집하기 위한 스크립트 이다.

```
CURDIR=`./solaris_2.7/pwd`
BINDIR="$CURDIR/solaris_2.7"
$BINDIR/echo "All output can be redirected to your choice of storage or netcat pipe."

$BINDIR/echo "========="
$BINDIR/echo "hostname:"
$BINDIR/echo "========="
$BINDIR/hostname
$BINDIR/echo
```

```
$BINDIR/echo "========="
$BINDIR/echo "disk use:"
$BINDIR/echo "========="
$BINDIR/df
$BINDIR/echo

$BINDIR/echo "==================="
$BINDIR/echo "disk format output:"
$BINDIR/echo "==================="
$BINDIR/echo "Need to add format output"
#fdisk -l
#need a binary for this
$BINDIR/echo

$BINDIR/echo "================================="
$BINDIR/echo "Local shell environment variables:"
$BINDIR/echo "================================="
$BINDIR/env
$BINDIR/echo

$BINDIR/echo "========================="
$BINDIR/echo "currently logged in users:"
$BINDIR/echo "========================="
$BINDIR/who
$BINDIR/echo

$BINDIR/echo "========================="
$BINDIR/echo "List of running processes:"
$BINDIR/echo "========================="
ps -efl
#$BINDIR/ps -efl ; need static ps command

$BINDIR/echo
$BINDIR/echo "======================"
$BINDIR/echo "Network interface info"
$BINDIR/echo "======================"
$BINDIR/ifconfig -a
$BINDIR/echo
$BINDIR/ifconfig -s
$BINDIR/echo
```

```
$BINDIR/echo
$BINDIR/echo "=================="
$BINDIR/echo "arp table entries:"
$BINDIR/echo "=================="
$BINDIR/arp -n
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "=================="
$BINDIR/echo "/etc/hosts file:"
$BINDIR/echo "=================="
$BINDIR/cat /etc/hosts
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "======================"
$BINDIR/echo "/etc/resolv.conf file:"
$BINDIR/echo "======================"
$BINDIR/cat /etc/resolv.conf
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "======================"
$BINDIR/echo "/etc/passwd file:"
$BINDIR/echo "======================"
$BINDIR/cat /etc/passwd
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "======================"
$BINDIR/echo "/etc/shadow file:"
$BINDIR/echo "======================"
$BINDIR/cat /etc/shadow
$BINDIR/echo

$BINDIR/echo
$BINDIR/echo "==================================="
$BINDIR/echo "netstat output(current connections)"
$BINDIR/echo "==================================="
```

```
$BINDIR/netstat -an
$BINDIR/echo

$BINDIR/echo "=============="
$BINDIR/echo "routing table:"
$BINDIR/echo "=============="
$BINDIR/netstat -rn
$BINDIR/echo

$BINDIR/echo "============================"
$BINDIR/echo "listening ports (via lsof):"
$BINDIR/echo "============================"
$BINDIR/lsof -P -i -n
$BINDIR/echo

$BINDIR/echo "============"
$BINDIR/echo "lsof output:"
$BINDIR/echo "============"
$BINDIR/lsof
$BINDIR/echo

$BINDIR/echo "==========="
$BINDIR/echo "Mount info:"
$BINDIR/echo "==========="
mount
$BINDIR/echo
$BINDIR/echo "==========="
$BINDIR/echo "    fstab    "
$BINDIR/echo "==========="
$BINDIR/cat /etc/vfstab
$BINDIR/echo
$BINDIR/echo

$BINDIR/echo "================="
$BINDIR/echo "SUID/SGID search:"
$BINDIR/echo "================="
$BINDIR/echo
find / -perm -2000 -o -perm -4000 -print | xargs $BINDIR/ls -l {}

$BINDIR/echo "================="
```

```
$BINDIR/echo "File permissions:"
$BINDIR/echo "================="
$BINDIR/echo "/etc:"
$BINDIR/echo "-----"
$BINDIR/ls -lrta /etc
$BINDIR/echo
$BINDIR/echo "/bin:"
$BINDIR/echo "-----"
$BINDIR/ls -lrta /bin
$BINDIR/echo
$BINDIR/echo "/sbin:"
$BINDIR/echo "------"
$BINDIR/ls -lrta /sbin
$BINDIR/echo
$BINDIR/echo "/usr:"
$BINDIR/echo "-----"
$BINDIR/ls -Rlrta /usr
$BINDIR/echo
$BINDIR/echo "/var:"
$BINDIR/echo "-----"
$BINDIR/ls -Rlrta /var
$BINDIR/echo
$BINDIR/echo "/dev:"
$BINDIR/echo "-----"
$BINDIR/ls -Rlrta /dev
$BINDIR/echo
$BINDIR/echo "/home:"
$BINDIR/echo "------"
$BINDIR/ls -Rlrta /home
$BINDIR/echo
$BINDIR/echo "/lib:"
$BINDIR/echo "------"
$BINDIR/ls -Rlrta /lib
$BINDIR/echo
$BINDIR/echo "============================================================"
$BINDIR/echo "================="
$BINDIR/echo "MD5SUMs:"
$BINDIR/echo "================="
$BINDIR/echo "/etc:"
$BINDIR/echo "-----"
```

```
$BINDIR/md5sum /etc/*
$BINDIR/echo
$BINDIR/echo "/bin:"
$BINDIR/echo "-----"
$BINDIR/md5sum /bin/*
$BINDIR/echo
$BINDIR/echo "/sbin:"
$BINDIR/echo "------"
$BINDIR/md5sum /sbin/*
$BINDIR/echo
$BINDIR/echo "/usr:"
$BINDIR/echo "-----"
$BINDIR/md5sum /usr/*
$BINDIR/echo
$BINDIR/echo "/var:"
$BINDIR/echo "-----"
$BINDIR/md5sum /var/*
$BINDIR/echo
$BINDIR/echo "============================================================"

#$BINDIR/echo "Make sure to grab /proc/kcore"
```

포렌식 분석

일단 Live 시스템에서 피해 시스템 분석을 끝냈고, 시스템을 포렌식 분석 단계로 넘어가야 한다면 Biatchux는 매우 유용한 도구이다. 또한 off-line 분석을 위해 시스템에서 하드 디스크 드라이브를 제거하기 어렵다면 Biatchux는 분석을 도와줄 뛰어난 도구이다. 아래에 설명한 단계들은 리눅스 및 윈도우에서 동일한 작업으로 진행하면 된다.

1. CD-ROM 으로 부팅하기
   -. 하드 디스크로 부팅하기 전에 CD-ROM이 먼저 부팅되도록 BIOS설정을 한다.
   -. Biatchux CD를 CD-ROM드라이브에 넣고 부팅을 진행한다.
2. 네트워크 설정
   -. CD-ROM으로 부팅이 진행되면 초록색 화면에서 1,3,4번 중 하나를 선택한다.
   -. 외부와 통신하기 위한 네트워크 인터페이스 설정을 한다(명령 행 입력은 3번 참조).
3. 디스크 이미지 복사
   -. Biatchux 메뉴를 통해 외부로 디스크 이미지를 비트 단위로 복사한다.
   -. 명령 쉘을 통한 비트단위 디스크 이미지 복사를 하려면 아래와 같이 진행하면 된다.

포렌식 시스템(피해 시스템에서 분석한 데이터를 저장할 시스템 혹은 저장장치)
Forensics 시스템 IP는 192.168.10.110 Forensics# nc –l –p 3147 > victim_sda1.img
Biatchux CD-ROM에서 실행
[root@FIRE] root> ifconfig eth0 192.168.10.111 netmask 255.255.255.0 broadcast 192.168.10.255 [root@FIRE] root> route add –net default gw 192.168.10.1 [root@FIRE] root> cd /data [root@FIRE] root>dd if=/dev/sda1 | nc –nv 192.168.10.110 3147

-. 위의 작업 과정이 오류 없이 끝났다면 피해 시스템의 디스크 이미지가 단일 파일(victim_sda1.img)로 포렌식 시스템으로 복사되었을 것이다.

4. 디스크 체크섬 비교 수행

-. 백업된 디스크 이미지와 원래의 디스크 이미지의 체크섬 값이 동일한지 비교한다. 만약 동일하지 않다면 다시 3번 과정을 거쳐 올바르게 백업된 디스크 이미지를 저장하여야 한다.

포렌식 시스템
Forensics# md5sum victim_sda1.img c8f201b2ffb65595a6c2af6a1c51bb66
Biatchux CD-ROM에서 실행
[root@FIRE] root>md5sum /dev/sda1 c8f201b2ffb65595a6c2af6a1c51bb66

5. 로컬 디스크 마운트 수행

-. 만약 적절한 포렌식 분석 절차를 따를 것이라면 시스템을 off-line상태로 둔 후 백업 받은 이미지를 가지고 분석해야 한다. 그렇지 않고 원래의 디스크 드라이브 이미지를 조사하거나 분석하려면 Biatchux로 부팅하여 피해 시스템을 조사하면 된다.

Biatchux CD-ROM에서 실행
[root@FIRE] root>mount –ro,noatime,nosuid,nodev,noexec /dev/sda1 /mnt/sda1 [root@FIRE] root> mount

6. SSHD(SecureShell Daemon)실행하기

: SSHD는 스크립트로 /sbin/에 저장되며, 환경파일은 /etc/ssh/sshd_config이다. 기본 접속 포트는 2222/tcp이다.

Biatchux CD-ROM에서 실행
[root@FIRE] root> start-sshd.sh [root@FIRE] root> netstat -ntl

```
[root@FIRE] sbin> netstat -ntl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
[           18:21:19]
[root@FIRE] sbin> ps -ef | grep sshd
root       606     1  0 18:05 ?        00:00:00 /usr/sbin/sshd -f /etc/ssh/sshd_
root       634     1  0 18:21 ?        00:00:00 /usr/sbin/sshd -f /etc/ssh/sshd_
[           18:21:23]
[root@FIRE] sbin> cat /sbin/start-sshd.sh
#!/bin/sh
echo "creating sshd keys"
ssh-keygen -b 1024 -f /etc/ssh/ssh_host_dsa_key -t dsa -P ""
ssh-keygen -b 1024 -f /etc/ssh/ssh_host_rsa_key -t rsa -P ""
#ssh-keygen -b 1024 -f /etc/ssh/ssh_host_key -t rsa1 -P ""

echo "Starting sshd server listening on port 2222"
/usr/sbin/sshd -f /etc/ssh/sshd_config &

[           18:22:02]
[root@FIRE] sbin>
```

7. VNCServer(vncserver)실행 하기

: VNC는 가상 네트워크 컴퓨팅(Virtual Network Computing)이다. 다양한 이기종 머신 으로부터 컴퓨터의 “데스크탑” 환경을 네트워크를 통해 볼 수 있는 원격 제어 프로그램 이다. VNC 서버가 실행중인 컴퓨터의 OS와 접속하려는 클라이언트의 OS환경이 같을 필요는 없고, JAVA를 기반으로 한 원격 웹 접속도 가능하다. VNC 홈페이지는 http://www.realvnc.com/ 이다. Biatchux에서 VNC서버를 실행하려면 아래와 같다.

Biatchux CD-ROM에서 실행
[root@FIRE] root> vncpasswd
[root@FIRE] root> vncserver :0 –geometry 1024x768 -depth 16
[root@FIRE] root> netstat -ntl

```
[root@FIRE] root> vncpasswd
Password:
Verify:
[           10:09:13]
[root@FIRE] root> vncserver :0 -geometry 1024x768 -depth 16

New 'X' desktop is prometheus:0

Starting applications specified in /home/root/.vnc/xstartup
Log file is /home/root/.vnc/prometheus:0.log

[           10:10:51]
[root@FIRE] root> netstat -ntl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:5800            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
[           10:10:58]
[root@FIRE] root>
```

▶ VNC를 통한 초기 접속 화면

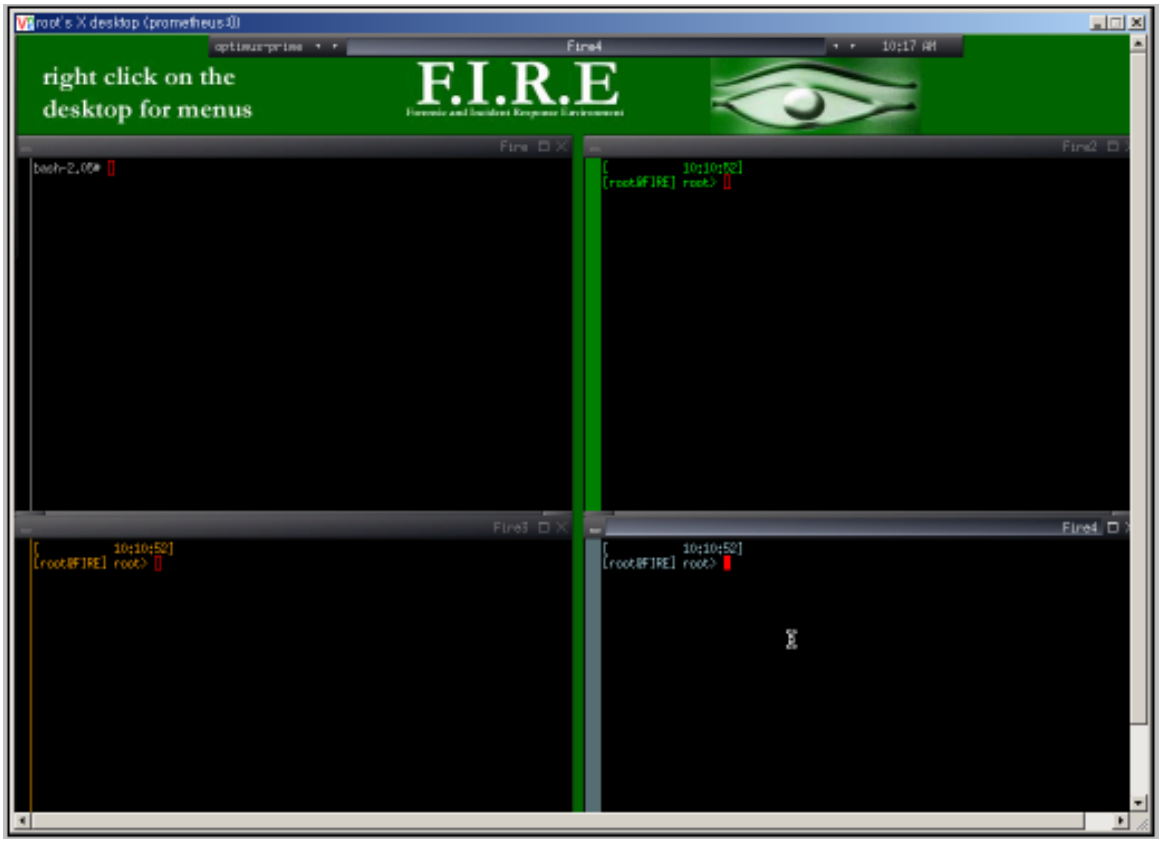


그림 9. Vncviewer로 접속한 Biatchux 0.4a의 X-Windows초기 화면

▶ Biatchux메뉴 구성도

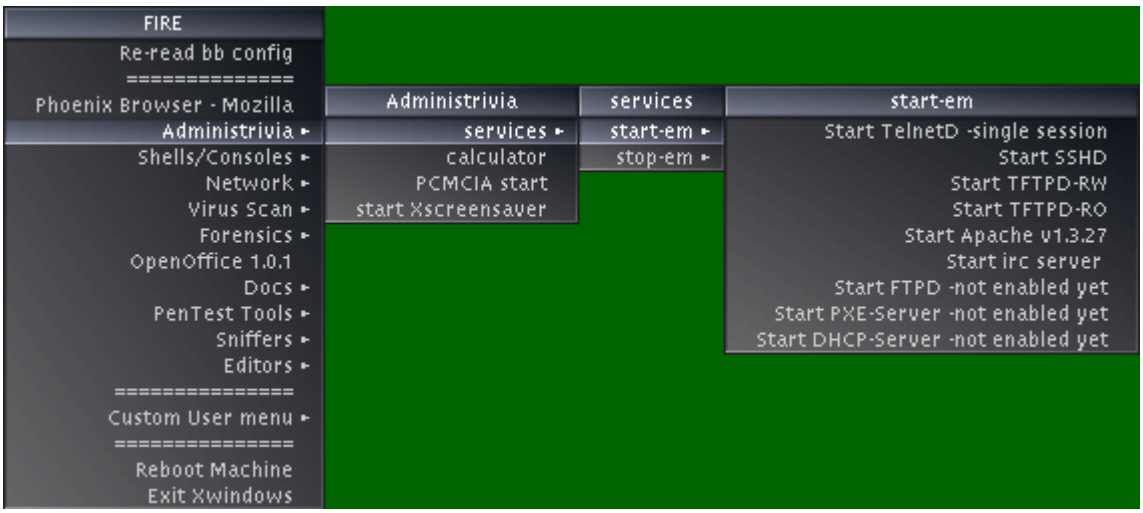


그림10. Biatchux 0.4a의 메인 메뉴

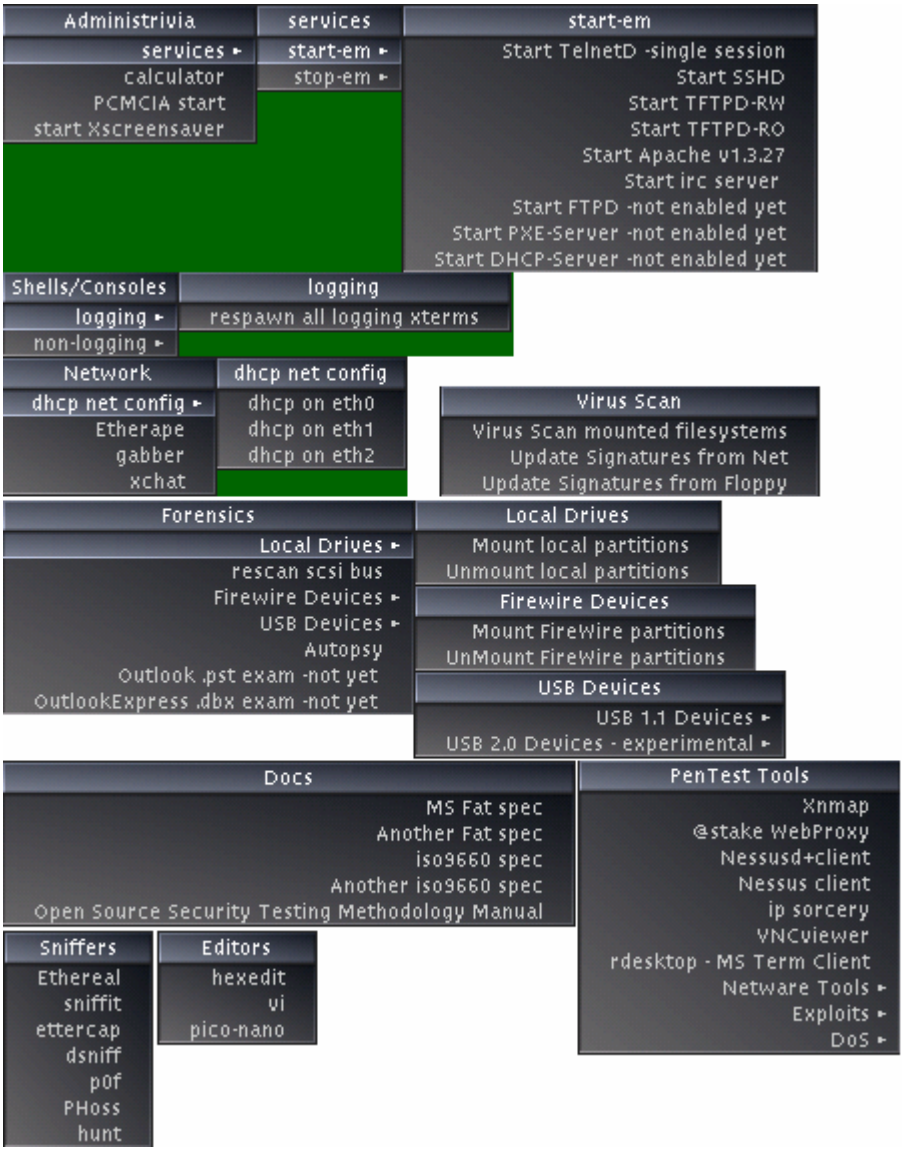


그림11. Biatchux 0.4a의 서브 메뉴들

그림12. Biatchux 0.4a에서 Mozilla 웹 브라우저 실행

디스크 이미지를 안전하게 복사하였다면 이제 Biatchux를 가지고 피해 시스템을 자세히 분석할 많은 유용한 도구를 사용할 수 있다. Biatchux는 find, grep등의 유닉스 표준 명령 이외에 유닉스 시스템의 포렌식 분석을 위한 특별한 툴을 포함하고 있다:

▶ chkrootkit – 루트킷의 설치 여부를 확인할 수 있는 툴이다.

▶ The Coroner's Toolkit(TCT) – 유닉스 시스템의 포렌식 분석을 수행하기 위한 툴들을 모아놓은 패키지.

▶ tctutils - Brian Carrier에 의해 작성되었으며, TCTUtils는 TCT를 확장한 유틸리티이다.

▶ Autopsy – TCT와 TCTUtils를 위해 GUI형태의 브라우저를 기반으로 이용하기 쉽도록 만든 것이다. 피해 시스템을 조사하는 사람이 포렌식 이미지를 파일, 블록, inode 레벨에서 분석하고 조사할 수 있다. 또한, 이미지 내에서 문자열 검색을 위한 편리한 인터페이스를 제공한다.

▶ mac-robber - 파일 이름과 관련된 것으로, 파일의 inode 변경(MAC), 수정, 접근시간을 연대순으로 timeline을 만들어 준다(grave-robber의 -m 옵션이다).

▶ stegdetect – 이미지 파일을 빠르게 자동적으로 찾아 주며 몇몇 감추어진 JPEG 파일의 숨겨진 정보도 볼 수 있다.

Examples: $ stegdetect *.jpg

cold_dvd.jpg : outguess(old)(***) jphide(*)

dscf0001.jpg : negative

dscf0002.jpg : jsteg(***)

dscf0003.jpg : jphide(***)

[…]

$ stegbreak -tj dscf0002.jpg

Loaded 1 files...

dscf0002.jpg : jsteg(wonderland)

Processed 1 files, found 1 embeddings.

Time: 36 seconds: Cracks: 324123, 8915 c/s

▶ hexedit – 디스크 이미지와 파일에서 hex 값을 편집할 수 있다.

▶ LDE – 리눅스 디스크 편집기 – inode 포맷 형태, 디렉터리 엔트리, 디스크 블록을 보고 편집할 수 있다.

▶ OpenOffice 1.0.1 – 오피스 프로그램.

Biatchux의 그 외 유용한 툴들

Biatchux는 침입자 분석 및 포렌식 이외에 다른 많은 기능을 수행할 수 있다. 그 외 유용한 툴은 아래와 같다:

▶ 취약점 분석: nmap, whisker, hping, firewalk, fragrouter, John-the-Ripper, nbtscan, nemesis, screamingCobra, onesixtyone, isnprober, hunt, p0f, THC-Hydra

▶ 네트워크 관리 및 보안 점검: ethereal, tcpdump, dig, dsniff, netcat, hunt, ncftp, ngrep, p0f, arena, Sniffit, Phoss, Wireless, airsnort, kismet, xchat, gabber

▶ 부하 테스트: hammerhead, apachebench

▶ 디스크/파일 복구: hexedit, LDE

▶ 침입탐지: snort

▶ 방화벽: ipchains, iptables

▶ 시스템 관리: telnetd, sshd, ftpd(차후 지원), tftpd, PXE-Server(차후 지원), Apache v1.3.27, irc server, DHCP-Server(차후 지원), 계산기, PCMCIA, X-screensaver

▶ 웹 브라우저: Mozilla

▶ 모의 해킹 툴: Xnmap, @stake WebProxy, Nessusd+client, ip sorcery, VNC Viewer, Netware Tools(Pandora), rdesktop(MS Term Client), Exploits(Unicode, jill, holygrail – solaris teleted), Denial Of Service Attack tool(SMBdie-kills NT/2k/XP))

▶ Exploit: bind관련, bsd관련, flood관련, holygrail, jill, local관련, rpc관련, sendmail관련, sparc관련, ssh관련, unencoded-stuff관련, win32관련, wuftp관련, random관련(기타 exploit)

▶ 편집기: hexedit, vi, pico-nano

Biatchux, PLAC, Trinux의 비교

	Trinux v. 0.80rc2	PLAC v. 2.9.5	Biatchux v. 0.4a
CD-ROM으로부터 부팅 가능한가?	○	○	○
floppy로부터 부팅 가능한가?	○		
배포 사이즈	1.44 MB*	47 MB	117 MB
시스템이 부팅한 이후에 미디어를 제거 할 수 있는가?	○	○**	X
부팅할 때에 모든 파티션을 자동적으로 검색하고 마운트 하는가?	X	○	X
명령을 메뉴 방식으로 지원하는가?	X	X	○
아래 유닉스 침해 사고 대응 및 포렌식 툴을 포함하고 있는가?			
TCT	○***	○	○
tctutils	○***	X	○
Autopsy browser	X	X	○
Ethereal	○	X	○
tcpdump	○	○	○
mac-robber	X	X	○
chkrootkit	X	○	○
LDE	○	○	○
hexedit	X	X	○
md5sum	○	○	○
netcat	○	○	○
StegDetect	X	X	○
아래와 같은 신뢰된 static binaries를 포함하고 있는가?	X	X	X
Windows	X	X	○
Linux x86	X	X	○
Solaris Sparc	X	X	○
윈도우 침해 사고 대응 툴을 포함하고 있는가?	X	X	○

Biatchux에 대한 질문과 답변

▶ F.I.R.E.에 관한 Q&A

Q: F.I.R.E. 은 무엇입니까?

A: F.I.R.E.는 부팅 가능한 CD-ROM위에서 침입대응 환경을 만들어 주는 프리웨어 포렌식 툴이다. 바꾸어 말하면 유용한 많은 보안 도구가 리눅스 배포판에 포함되어 쉽게 이용 가능하도록 만들었다. CD-ROM으로 부팅하므로 당신의 컴퓨터 위에서 아무것도 수정하지 않고 안전하게 시스템을 점검 할 수 있다. F.I.R.E.는 William Salusky에 의해 작성되고 유지된다. 더 자세한 사항은 홈페이지를 참조해라.

Q: F.I.R.E.가 할 수 있는 일은 무엇입니까?

A: 아래와 같은 많은 작업을 진행 할 수 있다.

- 피해 시스템으로부터 법률적 분석을 수행하고 손상된 데이터를 분석 수집할 수 있다.
- 신뢰된 바이너리를 사용하여 보안 사고를 신속히 대응할 수 있다.
- 손실된 파티션으로부터 데이터를 복구할 수 있다.
- 하드 디스크 드라이브를 깨끗한 환경으로 만들 수 있게 바이러스 체크가 가능하다.
- 취약점 분석이나 모의 해킹 테스트를 수행 할 수 있다.

F.I.R.E.는 X-Windows환경이나 표준 텍스트 콘솔로 운영할 수 있도록 부팅가능한 CD-ROM에서 수행된다. 양쪽 모두 동일한 결과를 만들어 준다.

Q: 이미 리눅스 배포판을 사용하고 있는 몇몇 보안/복구 프로그램과 비교하여 F.I.R.E.가 더 낳은 점은 무엇인가?

A: CD-ROM이나 플로피 디스크에 기반을 둔 몇몇 리눅스 배포판은 아래와 같다

- Knoopix
- Trinux
- PLAC

Knoopix는 어떤 하드웨어에서도 같은 방식으로 실행되는 리눅스이다. 전문적으로 하드웨어 탐지와 이에 관련된 응용프로그램에 대해 많은 양을 제공하지만 F.I.R.E.는 보안 전문가들을 위해 훨씬 더 많은 도구를 제공하고 있다.

Trunux는 매우 오래된 컴퓨터위에서 실행될 수 있도록 디스켓 3장으로 부팅가능하게 만들었다. 주로 무소음 라우터 프로젝트나 기타 특수한 경우에만 사용되고 있지만 F.I.R.E.는 그래픽 사용자 인터페이스 버젼의 소프트웨어를 실행할 수 있는 X-Windows를 기본적으로 내장하고 있고 훨씬 더 많은 유용한 도구들도 포함하고 있다.

PLAC는 CD-ROM으로 부팅가능한 리눅스 이다. 네트워크 감사, 디스크 복구, 포렌식 분석툴들을 포함하고 있다. 그러나 모든 메뉴가 명령행이나 스크립트로 이루어져 있어 사용하기 불편하다. F.I.R.E.는 메뉴방식을 도입하여 매우 쉽게 사용 가능하도록 만들었으며, 데이터 복구와 법률적인 분석을 전문적으로 할 수 있도록 개발되었다.

Q: 어떤 툴들이 포함되어 있는가?

A: F.I.R.E.에 포함되어 있는 툴들을 모두 서술하기에는 너무 많고, 몇몇 일반적인 툴들을 기술하면 아래와 같다.

- Nessus, Nmap, Whisker, hping2, hunt, fragrouter
- Ethereal, Snort, tcpdump, ettercap, dsniff, airsnort
- chkrootkit, F-Prot
- tct, tctutils, Autopsy
- Testdisk, fdisk, gpart
- SSH (client and Server), VNC (client and server)
- Mozilla, ircII, mc, Perl, biew, fenris, gpg

더 상세한 툴 목록은 웹사이트의 상세 목록표를 참고하여라.

Q: F.I.R.E.이 실행되기 위한 플렛폼은 무엇인가?

A: F.I.R.E.은 적어도 48MB램과 인텔 x86 호환 PC가 요구된다. X-Windows를 사용하기 위해서는 800X600 픽셀을 지원하는 베사 프레임 버퍼를 지원하는 모니터와 그래픽 카드를 필요로 한다. 물론 이때 마우스 또한 필요할 것이다.

Q: F.I.R.E.의 라이센스는 어떻게 되는가?

A: 자세한 사항은 상세한 라이센스 정책을 보면 된다. 기본적으로 당신이 사용할 목적으로 복사하여 사용하는 것은 가능하며, 판매 목적으로 사용해서는 안된다. 그리고 F.I.R.E.은 어떤 보증사항도 없이 배포된다는 것을 알아 두어야 한다.

▶ F.I.R.E.의 설치

Q: F.I.R.E.을 CD-ROM으로 어떻게 구할 수 있는가?

A: 현재는 F.I.R.E.은 이미지 파일(*.iso)로만 제공하고 있다. 인터넷 사이트로 부터 다운로드 해야하며, CD-ROM으로 구워서 사용하여야 한다(모뎀이나 ISDN으로 파일을 다운로드 하려면 회선속도에 따라 몇시간이상 소요될수도 있다. 이 이미지 파일은 대략 200MB보다 크다).

Q: OK, ISO이미지 파일을 다운로드 하였다. 지금 나는 무엇을 해야 하는가?

A: 1. Sourceforge에서 제공하는 체크섬값과 MD5 체크섬값을 비교하여 올바로 다운로드 되었는지 확인해라.
2. 파일의 무결성이 확인되면 iso파일을 CD-ROM으로 구워야 한다. 하나의 CD-ROM으로 올바로 구워졌는지 확인해라.
3. 당신의 컴퓨터의 BIOS 설정에 들어가 CD-ROM이 가장 먼저 부팅되도록 설정하여라.
4. F.I.R.E. CD를 CD-ROM드라이브에 넣은 후 컴퓨터를 부팅하면 수초 후에 녹색 화면의 환영 메시지를 보게 될 것이다. 텍스트 방식(3번)이나 그래픽 방식(4번)으로 부팅하면 된다.

Q: MD5 체크섬은 어떻게 계산하는가?

A: 유닉스 호스트 위에서 "md5sum" 프로그램을 사용하여 .iso 파일의 MD5 체크섬 값을 계산할 수 있다. 이

프로그램은 GNU 텍스트 유틸리티 패키지 안에 포함되어 있으며 대부분의 리눅스에 기본적으로 설치된다.

$ md5sum fire-0.4a.iso
ae810533dc3ae95e4036b2d665bd5f1a *fire-0.4a.iso

md5sum의 윈도우 명령 라인에서 사용 가능한 버전도 다운로드 할 수 있다. 만약 당신이 그래픽 사용자 인터페이스 버전에서 MD5 체크섬을 수행한다면 알고리즘은 DPASHA를 선택하고 수행하여야 한다.

예를 들면, 노트패드를 사용하여 다운로드된 파일과 체크섬 파일(예를 들면, fire-0.4a.iso.md5sum.txt)의 값을 서로 비교하여 그 값이 동일한지를 확인하여야 한다.

$cat fire-0.4a.iso.md5sum.txt(윈도우에서는 노트패드를 이용하여 보면 된다)
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

ae810533dc3ae95e4036b2d665bd5f1a *fire-0.4a.iso

-----BEGIN PGP SIGNATURE-----
Version: PGP 7.0.1

iQA/AwUBPearEQCUWsrxYo1REQKSdQCg5Dxsok4GFDLxZQchQs7q79TZLYcAn1I8
nNJ4BWGAfGsvpPOPsydl2HzQ
=iUvO
-----END PGP SIGNATURE-----

이 경우, 체크섬값은 ("ae810533dc3ae95e4036b2d665bd5f1a") 이며 다운로드한 파일과 동일하다면 파일의 무결성이 보장되는 것이다. 만약 다르다면 sourceforge의 미러링 사이트를 이용하여 다운로드를 재 시도해야 할 것이다.

Q: CD-ROM으로 이미지를 어떻게 구울 수 있는가?
A: iso파일을 CD-ROM으로 굽는 방법은 해당 CD-R/W 메뉴얼을 참고하길 바란다.

Q: F.I.R.E.의 root 패스워드는 무엇인가?
A: root 패스워드는 "firefire"이다.

Q: 나는 초기에 X-windows 시스템으로 부팅하였다. 어떻게 하면 콘솔 메뉴로 다시 시작할 수 있는가?
A: 단지 터미널에서 다음 명령을 입력하면 된다.

/sbin/dlg/startmenu

Q: 어떻게 하면 /data 디렉터리의 내용을 다른 곳으로 전송 할 수 있는가?

A: 컴퓨터가 네트워크로 연결되어 있어야 한다.
첫 번째로 네트워크 환경설정을 선택하여 외부와 통신할 수 있도록 만들어 주어야 한다.
그 후 원격 컴퓨터에서 netcat 프로세서를 시작하여 데이터를 수신할 수 있도록 만들어 준다:

```
[Shim@REMOTE]$ nc -l -p 3147 > data.tar
```

F.I.R.E. 시스템으로 돌아와서 데이터 디렉터리의 내용을 원격 컴퓨터로 전송한다(원격 컴퓨터의 IP Address는 192.168.1.100으로 예를 들었다):

```
[root@FIRE]# cd /data
[root@FIRE] tar cf - . | nc -nvv 192.168.1.100 3147
```

원격 컴퓨터에서는 tar 아카이브 파일을 압축해제 하여 데이터를 보관하면 된다.

```
[Shim@REMOTE]# tar -xvf data.tar
```

Q: 나는 부팅 후에 자동적으로 몇몇 프로그램을 시작하고 싶다.

A: "autoexec.sh"라고 불리는 쉘 스크립트를 만든 후 FAT로 포멧된 플로피 디스크의 루트 디렉터리에 위치시키면 된다.
만약 당신이 F.I.R.E.에서 이것을 수행하려면 아래와 같은 몇몇 명령을 입력하면 된다:

```
# nano autoexec.sh
# mcopy autoexec.sh a:
```

"autoexec.sh" 파일은 F.I.R.E.이 부팅하는 동안 찾도록 프로그래밍 되어 있다.
간단한 예를 들면:

```
---생략----
#!/bin/sh
insmod 8390
insmod ne2k-pci
ifconfig eth0 192.168.1.100 up
echo nameserver 192.168.1.254 > /etc/resolv.conf
loadkeys it
---생략---
```

당신이 만약 이미 F.I.R.E.를 수행중에 있다면 /sbin/checkfloppy.sh를 실행하면 된다.

Q: 어떻게 하면 F.I.R.E.을 현재 운용중인 시스템 위에서 사용할 수 있는가? 예를 들면, 내 PC를 부팅하지 않고 F.I.R.E.의 프로그램을 사용하고 싶다.

A: CD-ROM을 마운트한 상태에서 사용 가능하다. F.I.R.E.의 유용한 바이너리 파일들은 statically하게 연결되어 컴파일 되었으며, 만약 분석 시스템이 갖추어져 있다면 법률적인 분석에 바로 이용할 수 있다.

▶ F.I.R.E.의 트러블 슈팅

Q: 구워진 CD-ROM으로 부팅 할 수 없다.

A: 적어도 4가지의 경우일 수 있다.

1. 다운로드된 .iso파일이 손상되었다. MD5 체크섬값을 비교하여 파일의 무결성을 확인하여라.
2. CD-ROM이 잘못 구워졌다. iso파일에 문제가 없다면 다시 한번 CD-ROM을 구워야 한다.
3. CD-ROM으로 부팅 가능하도록 BIOS에서 설정을 하지 않았다. 부팅 시 BIOS설정에 들어가 CD-ROM이 가장 먼저 부팅되도록 설정하여야 한다.
4. F.I.R.E.이 요구하는 하드웨어 사양을 준수하지 못했다. 충분치 않은 RAM이나 x86호환 PC가 아니다.

Q: 내 컴퓨터는 매우 오래되어 CD-ROM으로 부팅할 수 없는 BIOS이다.

A: "Smart Boot Manager"를 플로피 디스크에 설치해라. 플로피 디스크로 부팅한 후 CD-ROM의 부트 매니져를 실행하여 부팅하면 된다. 만약 당신이 이러한 복잡한 과정을 거치기 싫다면, BIOS를 업데이트 하여 CD-ROM으로 부팅가능하도록 만들면 될 것이다. 대다수의 새로운 BIOS 버젼은 CD-ROM부팅을 지원한다.

Q: 나의 그래픽 카드는 i815 칩셋이다. x-Windows 시스템으로 부팅되지 않는다.

A: F.I.R.E.은 x11을 위해 VESA 프레임 버퍼 드라이버를 사용한다. i815 칩셋은 VESA 프레임 버퍼를 지원하지 않는다.

Q: 쉘이 이상하게 동작된다. 때때로 나는 아무것도 입력할 수 없게 되었다.

A: vi모드에서 Bash 환경을 설정하면 된다.

```
# set -o emacs
```

이것은 쉘을 Emacs방식으로 전환하게 된다. 매우 불편하게 될 것이다.

Q: 어떻게 하면 독일식 키보드 콘솔로 바꿀 수 있는가?

A: 아래 명령을 입력해라:

```
# loadkeys de
```

불행하게도 파일은 독일어를 지원하지 않는다. 다음 버전에 아마도 이를 지원할 것이다.
마찬가지로 이탈리어 키맵을 사용하려면:

loadkeys it

단지 키보드에서만 작업 가능할 것이다.

Q: 나는 NE2000 호환 PCI 이더넷 카드를 가지고 있다(예를들어 RealTek 8029 칩셋). 그러나 네트워크 설정을 할 수 없다.

A: 커널 모듈에 수동으로 로딩하여야 한다:

insmod 8390
insmod ne2k-pci

성공적으로 커널에 모듈이 올려지면 IP 환경을 설정하면 된다(DHCP나 고정IP로 설정).

Q: 나는 "autoexec.sh"파일을 만들었지만 자동적으로 실행되지 않는다.

A: 아래와 같이 메뉴얼로 실행하여라:

/sbin/chkfloppy.sh

Q: 버그를 발견한다면 어떻게 하면 되는가?

A: Sourceforge의 버그 트래킹을 방문하여 주길 바란다. 버그 트래킹에 기록된 것들은 다음 버전에서 수정될 것이다.

▶ F.I.R.E.의 그 외 사항

Q: 어디서 관련된 문서를 얻을 수 있는가?

A: Joe Lofshult가 FIRE의 문서를 작성하였으며, Biatchux라고 불려진다. Winword포맷으로 작성하였다. 더 상세한 정보는 F.I.R.E.의 Sourceforge 포럼을 방문하길 바란다.

Q: 또 다른 좋은 프로그램을 포함시키고 싶다.

A: 이미 포함되어 있는지 확인 한 후, Biatchux 프로젝트 홈페이지의 요구사항 페이지나 포럼에 기술해 주길 바란다.

참고 문서들(References)

■ @Stake. @stake Research Labs – Tools. <http://www.atstake.com/research/tools/index.html#forensic>.
■ Bajusz, Richard. Security Applications of Bootable Linux CD-ROMs. <http://rr.sans.org/linux/sec_apps.php>.
■ Dittrich, David. Basic Steps in Forensic Analysis of Unix Systems. <http://staff.washington.edu/dittrich/misc/forensics/>.

■ Dittrich, David. "Root Kits" and hiding files/directories/processes after a break-in. January 5, 2002. <http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq>.

■ Foundstone - Free Tools. <http://www.foundstone.com/knowledge/free_tools.html>.

■ Franz, Matthew. Trinux homepage. <http://trinux.sourceforge.net/>.

■ Lee, Rob. Incident-Response homepage. <http://incident-response.org/>.

■ Mandia, Kevin and Chris Prosise. Incident Response: Investigating Computer Crime. Berkeley: Osborne/McGraw Hill, 2001.

■ Project: Portable Linux Auditing CD. <http://sourceforge.net/projects/plac>.

■ Salusky, William and David Zendzian. Biatchux. <http://biatchux.dmzs.com/>.

■ Showalter, Brad. Trinux – A Digital Tool Belt. October 10, 2001. <http://rr.sans.org/unix/trinux.php>.

■ Somer, Lord. Linux Rootkit. <http://online.securityfocus.com/tools/1489>.

■ Vidstrom, Arne. Security Toolbox. <http://ntsecurity.nu/toolbox>.

■ Joe Lofshult < www.giac.org/practical/Joe_Lofshult_GSEC.doc >.

■ 심정재 “프리웨어 포렌식 툴킷” <http://www.securitymap.net/sp/docs/FreewareUnixForensicToolkit.pdf>.

■ 심정재 “윈도우 NT/2000 침입 대응” <http://www.securitymap.net/sp/docs/InitialResponseToWindowsNT_2000.pdf>.

이 문서는 포렌식 분석을 하기 위한 Biatchux 툴의 소개와 간단한 사용방법에 대한 글이다. 필자가 이용하는 프리웨어 포렌식 툴킷중의 하나로 2003년 5월 14일자에 0.35b에서 0.4a로 버전이 업그레이드 되었다. 몇몇 내용이 피치 못할 사정으로 빠져서 아쉽기만 하다. 처음부터 실제 피해 시스템의 디스크 이미지를 가지고 Biatchux를 이용한 분석방법에 대해 소개하였지만, 외부적인 문제(SRC IP노출, DST IP노출, 디스크 덤프이미지 제공, 해킹 시연과정 등등)로 인하여 상당한 부분이 삭제가 되었다(—.—;). 현재 이미지내의 IP와 관련된 중요한 정보를 수정하여 차후에 공개할 예정이다.

해킹 사고는 더 이상 남의 일이 아닌 일반 시스템, 네트워크 관리자가 경험하게 되는 평범한 현상이다. 하지만 대부분의 시스템/네트워크 관리자들은 이러한 인터넷 공격 및 공격자에 대한 지식, 경험 부족으로 인하여 피해 시스템에 대한 제대로 된 분석을 하지 못하고, 결국 반복적인 침입을 당하는 결과를 가져온다. 아무쪼록 이 문서가 이러한 행위들로부터 자신의 시스템을 보호하고 해킹사고가 국내에 확산되는 것을 방지하였으면 하는 바램이다.

이 문서를 타 사이트로 이전 시에는 반드시 www.securitymap.net과 필자를 기술해 주길 바란다.

이 문서의 궁금한 사항이나 기타 문의는 아래 E-Mail로 질문해 주길 바란다.

jjshim AT hanaro.com 심정재
