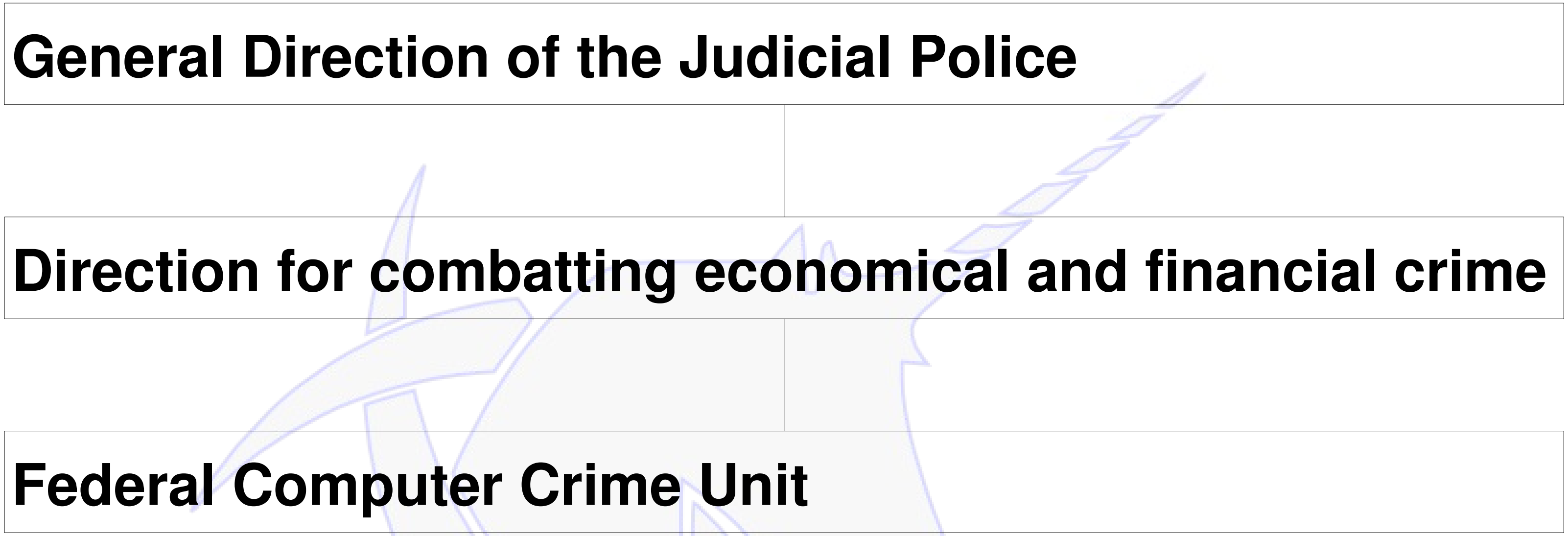
FCCU GNU/Linux Forensic Boot CD

Hack.lu Forensic Workshop

Christophe Monniez
Geert Van Acker

Who we are ...



General Direction of the Judicial Police

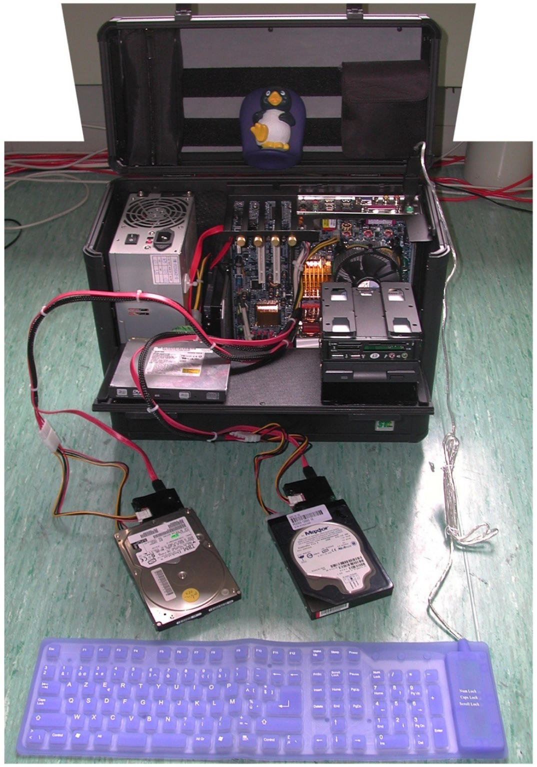
Direction for combatting economical and financial crime

Federal Computer Crime Unit

- Federal Police structured on two levels
- Every district has a “Regional Computer Crime Unit
  - Assistance housesearches
  - Forensic analysis ICT
  - Internet investigations

Flight case ?

- Intervention kit FCCU
- ATA, SATA, FireWire, USB, Cardreader, DVD, ...
- Distribute evidence for this workshop



www.d-fence.be
www.lnx4n6.be

FCCU GNU/Linux Forensic Boot CD

CD presentation

- Primary goal is to make forensic images
  - Disk 2 Disk
  - Partition 2 Partition
  - Disk/partition 2 file

CD presentation

- Difference with other non forensic boot cd
  - No automatic use of swap partitions
  - Lots of forensic tools
  - Doesn't start in graphical mode
  - No daemons at startup
  - Custom kernel with good usb support (8.1 & 9.0)
  - Frequently updated
  - Belgian keyboard by default
  - all FCCU scripts/progs begin by “fccu”

CD Goals

- Determine if PC is suspect (speed)
  - Low level search
    - keyword search
    - salvage based on file signatures
  - Childporn
    - Linux frame buffer image viewer (http://linux.bytesex.org/fbida/)
    - mplayer with framebuffer support
  - Internet traces
    - Mozilla – Firefox
    - Internet Explorer

The evidence

- Presentation of The Evidence
  - A 128 MiB USB key
  - Suspect traces are
    - named "forensic target ..."
    - everithing about "pirates"

The evidence

- Forensic sound imaging
- The tools :
  - dd
  - sdd
  - dcfldd
  - dd_rescue
  - GNU ddrescue
  - dd_rhelp

The evidence

- Through the network using Netcat & dd:
  - Suspect PC:

```
#dd if=/dev/sda conv=noerrors,sync | pipebench | netcat -l -p 2000
```

  - Trusted PC:

```
#netcat 192.168.x.x 200x | pipebench > /mnt/forensic/usbkey.dd
```

```
#netcat 192.168.x.x 200x | pv -i 1 -s 128m > /mnt/forensic/usbkey.dd
```

Tips

- Compression is you friend !
  - Suspect PC :

```
#dd if=/dev/sda conv=noerrors,sync | pipebench | gzip -fast | netcat -l -p 2000
```

  - Clients :

```
#netcat 192.168.x.x 200x | gunzip | pipebench > /mnt/forensic/usbkey.dd
```

```
#netcat 192.168.x.x 200x | gunzip | pv -i 1 -s 128m > /mnt/forensic/usbkey
```

The evidence

- Identifying devices (goals)
  - you have to know what to copy
  - writing an accurate report
  - finding suspicious informations

The evidence

- Identifying devices
  - general informations

```
# cat /proc/partitions
# lshw
# cat /proc/mem
# cat /proc/cpuinfo
# dmesg
```

The evidence

- Identifying devices
  - ATA/IDE
    - Try to find serial numbers
    - name your image using the serial number

```
# ide_info /dev/hdx
# lshw
# hdparm -i /dev/hda
# hdparm -I /dev/hda
```

The evidence

- Identifying devices
  - HPA/DCO

```
# dmesg (maybe kernel 2.6.10 only)
# hdparm -I /dev/hdx
# disk_stat /dev/hdx
```

  - USB/FireWire/SATA

```
# cat /proc/scsi/scsi
# scsiinfo -s /dev/sda
```

Tips

- Redirect into information file(s)

```
# lshw >> usbkey-info.txt
```

- Never forget bash automatic completion
- Read man pages

The evidence

- Imaging verification

```
# md5sum usbkey.dd
```

```
# md5sum /dev/sda
```

```
# sha1sum usbkey.dd
```

```
# sha1sum /dev/hda
```

Tips

- Think like a plumber !
  - Why not using tee to calculate the hash during the imaging

```
#dd if=/dev/sda | tee usbkey.dd | md5sum > usbkey.md5
```

  - The same with a progress bar

```
#dd if=/dev/sda | pipebench | tee usbkey.dd | md5sum > usbkey.md5
```

The evidence

- Once imaging is done, try to identify filesystems
  - DOS type partioning

```
# fdisk -lu usbkey.dd
# sfdisk -luS usbkey.dd
```

  - Other types
    - DOS type
    - MAC type
    - BSD disklabels
    - SUN

```
# mmls usbkey.dd
```

The evidence

- Is it really a partition magic recovery partition ?

```
# disktype usbkey.dd
```

- disktype recognize and probes partition types
  - DOS
  - APPLE
  - AMIGA
  - ATARI ST
  - BSD Disklabels
  - Linux Raid, LVM 1 & 2
  - Solaris (x86 & sparc)

The evidence

- Mounting the filesystem read-only

```
# insmod /lib/modules/2.6.11/kernel/drivers/block/loop.ko.distrib
# mount usbkey /mnt/forensic -o loop,offset=$((51*512)) -r
```

ATTENTION JOURNALING FILESYSTEM

The evidence

- Basic informations about the filesystem
  - Counting regular files

```
# find /mnt/forensic/ -type f | wc -l
```

  - Partition usage

```
# df -h /mnt/forensic/
```

Keyword search

- Choosing the right keywords is the most difficult part
- What are we searching for ?

– “*Wolves of the sea*” by Randall Parrish	“You are pirates”
– “*In Search of the Castaways*” by Jules Verne	“pirates! pirates”
– “*The Prince*” by Nicolo Machiavelli	“fearing no evil”
– “*CryptonomiconCypherFAQ*”	“pizza delivery driver”
– The Doors song “*the end*”	“the end of everything”

Keyword search

- Choose a text editors
  - vim
  - mcedit
  - echo “You are pirates” >> keywords.txt
- Create a text file named keywords.txt with those keywords : (NO EMPTY LINES)
  - You are pirates
  - pirates! pirates
  - fearing no evil
  - pizza delivery driver
  - the end of everything

Low level Keyword search

- The simple way :

```
# cat usbkey | strings | egrep -i -f keywords.txt
```

Finding the position on the image

```
# cat usbkey | strings -td | egrep -i -f keywords.txt
# cat usbkey | strings -tx | egrep -i -f keywords.txt
```

Adding colors

```
# cat usbkey | strings -td | egrep --color -i -f keywords.txt
# cat usbkey | strings -td | glark -N -i -f keywords.txt
(slower but “glark” works with “| less -r”)
```

Viewing more context

```
# cat usbkey | strings -td | egrep -5 --color -i -f keywords.txt
```

Low level Keyword search

- Don't forget other encodings
  - 16 bits little endian

```
# cat usbkey | strings -td -el | egrep --color -i -f keywords.txt
```

  - 16 bits big endian

```
# cat usbkey | strings -td -eb | glark -N -i -f keywords.txt (slower)
```

- Possibility to do all in one pass
  - Think like a plumber !
  - usage of “mkfifo”
  - usage of “tee”

Low level Keyword search

- Extracting fragments of results
  - “You are pirates” was found at offset 15393432

```
#dd if=usbkey.dd skip=$((15393432/512)) count=1 | strings
```

- Use redirection to save in files
- Save in files without filtering with strings
- Scripting possibilities

Low level Keyword search

- That's great but I want to know if the result is in a file or not !
  - Usage of sleuthkit
  - “You are pirates” was found at offset 15393432
  - “ifind“ : a sleuthkit tool to find information about a disk unit
  - “istat” : a tool to display details of an inode

```
# ifind -o 51 -d $((15393432/512)) usbkey.dd
```

- The inode “1397-128-4” is returned

```
# istat -o 51 usbkey.dd 1397 “1397-128-4” | less
```

```
# istat -o 51 usbkey.dd 1397 “1397-128-4” | egrep “^Name:”
```

Low level Keyword search

- Now try it with the other results

```
# ifind -o 51 -d $((26619086/512)) usbkey.dd
```

1469-128-4

```
# istat -o 51 usbkey.dd 1397 “1469-128-4” | less
```

Low level Keyword search

- Let's continue

```
# ifind -o 51 -d $((39473367/512)) usbkey.dd
```

- Inode “1476-128-4”

```
# istat -o 51 usbkey.dd “1476-128-4” | egrep “^Name”
```

```
In_Search_of_the_Castaways_by_Jule
  s_verne.doc
```

Low level Keyword search

- The last one

```
# ifind -o 51 -d $((41624592/512)) usbkey.dd
```

- Inode “1478-128-4”

```
#istat -o 51usbkey.dd 1478 | egrep “^Name”
```

```
65544bytes-doc.txt
```

Low level Keyword search

- Finding the files on the mounted filesystem
  - *Wolves_of_the_sea.doc*

```
#find /mnt/forensic/ -iname “wolves*”
```

  - *Did you find it ?*
  - *Let's verify with a keyword search against the file*

```
# cat “/mnt/forensic/Documents and Settings/Rackham/My Documents/\
Wolves_of_the_sea.doc” | strings | \
egrep -i --color -f /tmp/keywords.txt
```

MS WORD files Tip

- Viewing an MS Word file

```
# cd “/mnt/forensic/Documents and Settings/Rackham/My Documents/”
# wvText “Wolves_of_the_sea.doc” /tmp/wolves.txt
# less /tmp/wolves.txt
```

- Try “wv[TABTAB]”
- wv even support protected MS Word files (you must know the password :-))

```
# antiword “Wolves_of_the_sea.doc”
```

```
# catdoc “Wolves_of_the_sea.doc”
```

MS WORD files Tip

- Obtaining info about MS Word file

```
# wvSummary Wolves_of_the_sea.doc
```

```
# wvVersion Wolves_of_the_sea.doc
```

- Will give info about encryption

```
# find /mnt/forensic/ -iname “*.doc” -exec wvVersion '{}' ';'\
 | egrep -v “Encrypted: No”
```

- Will find all encrypted “.doc”

```
# fccu-docprop Wolves_of_the_sea.doc 2>/dev/null
```

- Usefull informations about dates (last print ...)
- works even better with “xls” files

Low level Keyword search

- Finding the files on the mounted filesystem
  - *In_Search_of_the_Castaways_by_Jules_verne.doc*

```
# find /mnt/forensic/ -iname “in_search*”
```

  - *Did you find it ?*

Low level Keyword search

- Finding the files on the mounted filesystem
  - *65544bytes-doc.txt*

```
#find /mnt/forensic/ -iname “65544bytes*”
#cat 65554bytes-doc.txt | strings | egrep -i --color -f /tmp/keywords.txt
```

Low level Keyword search

- Trying to find answers
- Extracting the unallocated space using The Sleuthkit

```
# dls /dev/loop0 > /tmp/unallocated.dd
```

- Search for keywords in the unallocated space

```
# cat /tmp/unallocated.dd | strings | egrep -i -f /tmp/keywords --color
```

- First answer found !
- Two of the texts are in the unallocated space
- There is a good chance that they may be deleted files

Low level Keyword search

- Trying to find answers
- Extracting the slackspace using The Sleuthkit

```
# dls -s /dev/loop0 > /tmp/slackspace.dd
```

- Search for keywords in the slackspace

```
# cat /tmp/slackspace.dd | strings | egrep -i -f /tmp/keywords --color
```

- Bingo !
- The Doors lyrics are in the slackspace of the file “65544bytes-doc.txt”

Deleted files

- What kind of filesystem ?
  - NTFS -> ntfsundelete
  - FAT -> The Sleuthkit
  - ext2 -> The Sleuthkit

Deleted files

- Finding deleted files
  - In this case, we use /dev/loop0

```
# ntfsundelete /dev/loop0
```

Wow

```
# mkdir /tmp/recovered
# ntfsundelete /dev/loop0 -u1388 -d /tmp/recovered
# fbi /tmp/recovered/forensic-target-1.jpg
```

Wow again !

Deleted files

- Finding deleted files

```
# ntfsundelete /dev/loop0 -u1471 -d /tmp/recovered
# mplayer /tmp/recovered/forensic-target-2.mpeg
```

```
    Another pirate caught !
```

- Scripting

```
# ntfsundelete /dev/loop0 -p 100 | awk '{ print $1 }' |\
 egrep  “^[[:digit:]]” | while read inode ;\
 do ntfsundelete /dev/loop0 -u${inode} -d /tmp/recovered/ ; done
```

```
    All done !
```

Fighting Childpr0n

- Search/view pictures using “fbi”
- On the mounted filesystem

```
# find /mnt/forensic/ -iname “*.jpg” -exec fbi -a '{}' ';'
```

- Use all the power of find

```
# find /mnt/forensic/ -iname “*.jpg” -size +100k -exec fbi -a '{}' ';'
```

Fighting Childpr0n

- Search/view movies using “mplayer”
- On the mounted filesystem

```
# find /mnt/forensic/ -iname “*.mp*” -exec mplayer -ao null '{}' ';'
```

Fighting Childpr0n

- File salvage based on header-footer
- magicrescue
  - Create output directory (can be hughe !)
  - Use it on unallocated space to maximize your chances
  - Recipes are in “/usr/share/magicrecue/recipes”

```
# mkdir /tmp/rescued
# dls /dev/loop0 > /tmp/unallocated.dd
# magicrescue -r /usr/share/magicrescue/recipes/jpeg-jfif -r \
 /usr/share/magicrescue/recipes/jpeg-exif \
 -d /tmp/rescued/ /tmp/unallocated.dd
# fbi /tmp/rescued/*
```

press “i” to view exif informations

Fighting Childpr0n

- Lot of progs to view meta informations in files

```
#extract -f /tmp/rescued/*
```

```
#exif /tmp/rescued/*
```

```
#exiftags /tmp/rescued/*
```

```
#jhead /tmp/rescued/*
```

- U can use dupemap and magicsort
- to remove duplicates
- to sort files

Fighting Childpr0n

- foremost
  - Copy and adapt the config file

```
# cp /etc/foremost.conf /tmp/
# vim /tmp/foremost.com
```

- uncomment all “jpg” lines
- Create an empty directory

```
# mkdir /tmp/fresult
# foremost /tmp/unallocated.dd -o /tmp/fresult -c /tmp/foremost.conf
```

The Way Of The Exploding File

- Is there compressed “zip” files on the system ?

```
#find /mnt/forensic -type f -iname “*.zip”
```

- Maybe a zipped file but without a zip extension

```
#find /mnt/forensic -type f -exec file '{}' ';' | egrep “Zip”
```

- “/mnt/forensic/tempfiles/thisisnotapipe.dll” ???

```
# unzip -l /mnt/forensic/tempfiles/thisisnotapipe.dll
```

```
    Oh Oh
```

The Way Of The Exploding File

```
# unzip /mnt/forensic/tempfiles/thisisnotapipe.dll -d /tmp/
```

- Ooops, password protected

```
#fcrackzip -D -p /usr/share/dict/french -u \
/mnt/forensic/tempfiles/thisisnotapipe.dll
```

```
# unzip /mnt/forensic/tempfiles/thisisnotapipe.dll -d /tmp/
Enter the password you found
```

- All done !

NTFS Alternate Data Streams

- Finding NTFS ADS

```
#fls -r /dev/loop0 | sed “s/:/;/” | egrep “:”
```

```
#ffind /dev/loop0 1470
```

```
#ffind /dev/loop0 1470-128-5
```

```
#icat /dev/loop0 1470-128-5 > /tmp/borderline.dat
#file /tmp/borderline.dat
```

```
#mplayer -ao null /tmp/borderline.dat
```

NTFS Compressed folders

- Natively supported by the GNU/Linux NTFS driver
- Low level search seems compromised !
  - remember your keyword search for “*The Prince*”
  - The keywords were “fearing no evil”
  - they were found in unalocated space
- Try

```
#find /mnt/forensic -iname “*.txt” \
-exec egrep -H -i --color -f /tmp/keywords.txt
```

```
#fls -r /dev/loop0 | egrep -i "theprince.txt"
```

```
#istat /dev/loop0 1469 | less
```

MS WINDOWS leave traces !

NTFS encrypted folders

- Filenames are visible

```
# cd “/mnt/forensic/Documents and Settings/Rackham/My Documents/"
# ls Kryptonite
```

```
# cat “Kryptonite/CryptonomiconCyherFAQ.html”
```

```
# fls -r /dev/loop0 | egrep -i "kryptonite"
# istat /dev/loop0 1472 | less
```

```
# fls -r /dev/loop0 | egrep -i "cypherfaq"
# istat /dev/loop0 1474 | less
# icat /dev/loop0 1474
```

- Start the Evil OS

Timeline filesystem

- Extract MAC times files:

```
# fls -o 51 -m "E:" -r usbkey.dd > /tmp/body
```

  - ils for deleted files

```
# ils -o 51 -m usbkey.dd >> /tmp/body
```

- Presentation:

```
# mactime -d -b /tmp/body 9/08/2005-10/23/2005 | less
```

Web surf traces

- Internet Explorer activity forensic

```
# find /mnt/forensic -iname "index.dat" -exec pasco '{}' ';'
```

- Webmail ?

```
# find /mnt/forensic -iname "index.dat" -exec pasco '{}' ';'\
| egrep -i --color mail
```

- Any password ?

```
# find /mnt/forensic -iname "index.dat" -exec pasco '{}' ';'\
| egrep -i --color pass
```

Web surf traces

- Google searches ?

```
# find /mnt/forensic -iname "index.dat" -exec pasco '{}' ';'\
| egrep -i --color "search\?"
```

- Terrorism interest ?

```
# find /mnt/forensic -iname "index.dat" -exec pasco '{}' ';'\
| egrep -i --color "bomb"
```

Web surf traces

- Mozilla / Firefox history

```
# find /mnt/forensic -iname "history.dat" -exec mork.pl '{}' ';'
```

  - Export to html is possible

Event log files

- Search for EVT files and parse them

```
# find /mnt/forensic -iname "*.evt" -exec fccu.evtreader.pl '{}' ';'
```

  - Export to html is possible
  - May not be complete
  - May help to discover useful events like removable devices
  - may help in timelining
  - more complete tools on the next CD version

Clamscan

- Finding viruses on the mounted filesystem

```
#clamscan -i -r /mnt/forensic
```

- You can use a previously downloaded virus database

```
#clamscan - i -r -d /tmp/mydatabase /mnt/forensic
```

Future

- PXE boot for automatic keyword search in multiple machines
- more tools
  - grokevt
  - reglookup
  - who knows ?

What we didn't talk about

- Many tools that are on the CD
  - See index.html in the root of the CD
- Brief overview :
  - SAM files forensic analysis with "*chntpw*"
  - PST files conversion with "*readpst*"
  - "*Ftimes*" a tool to help in :
    - keyword search
    - alternate streams / hidden files search
    - timeline
  - Network tools
  - Pen testing tools
  - Password cracking tools

That's all

Thank you for your attention