Kudo Daido Juku allows the fighter to adapt to each new situation using the Budo spirit as their guide.

This tutorial/workshop was developed by, Sandro Melo – Bandtec College (sandro.melo@bandtec.com.br) – 4NIX (sandro@4nix.com.br), with the goal to be a reference in the studies of the Computer Forensic Course, using many FLOSS tools (Free/Livre and Open Source Software).

About me

About Sandro Melo – aka CARACOA - Currently I work at Bandtec College, and also with Advanced Training, Forensics Response to Security Incidents and Computer Forensic and student/candidate in Doctor Program in TESOR-PUC-SP - I was born in the beautiful city of São Paulo, Brazil. I moved to São Paulo where I began my professional career in System Security. Since 1996 I have worked mainly with Linux/ FreeBSD and FLOSS (Free/Livre and Open Source Software), Network Administrator. I am often a guest professor at many universities all over Brazil. Project Fedora Linux Ambassador, LM and RSA FRACTOR.

I take great pride in everything I do, especially with my work in Forensics. I have years of hands-on experience with many of the core technologies and have written many books and articles on security and forensics. When not working or writing, I can be found experimenting with the latest Open Source solutions, including new versions of the same Operation Systems like Unix, such as Linux, FreeBSD or Mac OS X and also some FLOSS tools because I find it enjoyable and have a deep passion for my work.

"Hi been over bij u in BruCON / Fa trots heus eu lid dans BruCON"
This workshop was developed by myself, with the goal of being referred to in the study of the Computer Forensic Course, using many tools of F.L.O.S.S. FLOSS means (Free/Libre and Open Source Software).

Introduction

In the past, servers configured their risks but these risks were physically dimensioned, corresponding to the limits of the LAN of the corporation or institution. The Internet has radically changed this scenario.

It is more secure than a system with Firewall or other security devices, there will always be the possibility of human error or hitherto unknown failure in the operating system or applications, whether proprietary or FLOSS system. Given this degree of risk, or in fact intangible, the threat of an invasion is something that we can't overlook.

In this context, forensic techniques are essential during the response to an incident, as to identify where the computer system was compromised, and what information was stolen or changed, also to identify the attacker and preparing the environment for the expertise of Computer Forensics.

Bearing in mind the care of an expert in Computer Forensics, the intrusion system is an electronic crime. Digital evidence must be preserved so that it can be of value.

In the past, maybe 30 years ago, we had standalone computer systems, Nowadays everything is connected to the Internet

This brings new possibilities/ but also new and bigger problems

“Initial Concepts”

Post Mortem Forensic

Live Forensic

Network Forensic
We can divide the process into 3 phases.

**Live Forensics** covers the actions at the moment data is collected, all information about the system state, when the computer and your devices are turned on.

**Network Forensics** is related to network traffic of the computer with other computers and appliances. It contains information about mail server, proxy server, web server, IPS, IDS and firewall logs.

These 2 phases consist of a collection of detailed information that can be very useful in **Post Mortem Analysis**.

**Post Mortem analysis** is a very important moment, where the expert has to analyze the collected media images.

Now there is a very important stage, that is, **"Memory Forensic"**, where the expert can also do a "Post Mortem" of memory information using the "memory dumping" collected during **Live Forensics**.

**Image for Post Mortem**

Practically the whole Post Mortem begins when the file image is created. The image can be created in specific formats such as:

- **RAW** - created with some command like dd such as dd3cd. Typical format used in systems like Unix for any filesystem (NTFS, FAT, EXT3, UFS)

- **Librew** - Default format of Encase tools and supported for Linux with command libblabla

There are specific formats to create a **image forensic**

There are specific formats to create a **image forensic**
The Live Forensic represents data collection while the device is turned on. This information shows a certain degree of volatility that must be considered during the process.

The expert should perform the collection process starting from the most volatile to the least. This slide refers to what is suggested in RFC3227, the expert must know that the information in the registry and cache is extremely volatile. For this reason the collection becomes irrelevant, because the simple act of starting the registry and cache collection changes their state.

Network forensics can provide interesting data, and even simple clues can be very useful during Post Mortem Forensics. It is a fact that the better the network structure and the more security assets the network has, the higher the quality of data collection.

For example: If you have a CCTV system but with bad images, or if you don’t have CCTV at all, how can you know who entered your home or business? So, if an incident occurs in a network without security assets, such as firewall, proxy, IDS, IPS, logs server, you will need to collect more information because if we only have a gateway, little or no relevant information can be collected in this phase.

Network Forensics can provide interesting data, and even simple clues can be very useful during Post Mortem Forensics. It is a fact that the better the network structure and the more security assets the network has, the higher the quality of data collection.

For example: If you have a CCTV system but with bad images, or if you don’t have CCTV at all, how can you know who entered your home or business? So, if an incident occurs in a network without security assets, such as firewall, proxy, IDS, IPS, logs server, you will need to collect more information because if we only have a gateway, little or no relevant information can be collected in this phase.
During the Post Mortem the expert has lots of work to do, because he needs to analyze a great amount of data, attempting to find clues and possible evidence.

The Post Mortem Analysis process consists of image data in the media, related to the compromised computer.

The correlation of the collected information during Live, Memory and Network forensic, is exciting. Why?

Because most of the time, all the information coming together, creates a clearer picture. The proposal of this workshop is to suggest that analysis of the media image, related to the incident, can be done by using the concept of 5-layer methodology.

It must also be considered that during the Post Mortem, the expert can find files that demand more detailed analysis.

These files are called artifacts. Artifact analysis is an interesting stage that demands special care, depending on its type. But that is not the subject of this workshop.

Correlating the information collected during Live, Network, Memory with Post Mortem Analysis is the most important action, so I recommend you do this as much as possible.

That is why the information should be correlated during the incident analysis. As it can help to identify relevant evidence.

Let’s remember that in some cases, there will only be media images to analyze, which doesn’t make an investigation impossible.

It is a fact that with increasing storage capacity, Post Mortem analysis becomes more and more difficult, because of the amount of data that needs to be analyzed.
Initial System Analysis

Several actions can be taken in an attempt to find evidence and artifacts related to Security Incidents under investigation.

Knowing the “bad guy’s” Modus Operandi helps the Computer Forensic Expert to do his/her job. However, unusual and stealth behavior will always present a challenge.

“Bad guys” who do not have advanced technical knowledge have a Modus Operandi that usually leaves behind evidence of their actions.

Byte Map creation

The creation of an Image String file, as a first step, may allow the identification of relevant information.

# strings -a image.img | tee image.img.strings

The strings command has support only ASCII format, that why we need to get other different type of strings, use the srch_strings command:

# srch_strings -a image.img | tee image.img.strings
The String extraction process can be executed at various stages of the forensic analysis.

It is a good idea to do this even before the 5-layer analysis is done, so that the possible clues can help identify how the incident happened.

These clues can also make the identification of evidence possible.

**Strings vs Regex**

The use of REGEX when dealing with string files is an essential mechanism. This way, the use of tools like GREP, EGREP, GLARK are useful to extract clues.

```
# grep -i "tar.gz" image.string
# egrep --regexp=".tgz|.zip|.bz2|.rar|.c" image.string
# grep -E "[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}" image.string
```

At this point, the correlation of information about all forensic phases and also information about strings collected from the media is very interesting.

**Strings vs Regex**

```
# grep -i "\/exploit/" imagem.string
# grep -i "\/exploits/" imagem.string
# grep -i "\rootkit/" imagem.string
# grep -i "\./\./\/" imagem.string
```

Here you can see examples of grep and egrep commands with specific regex

```
grep -i "\/bk/" image.string
grep -i "xpl" image.string
grep -i "force" image.string
grep "\./\./\./" image.string
grep "SSH_CLIENT=" image.string
```
Here there are some other examples of grep command:

```bash
# cat image.img.strings | grep -i -f arq.txt

# cat image.img.strings | egrep -i -color -f arq.txt

# cat image.img.strings | grark -N -i -f arq.txt
```

Knowing that, it is necessary for the expert to use the commonly known REG-EX dictionary and create smart REG-EX based on knowledge of the incident using key words. All the tools of REG-EX allow the use of a dictionary file with interesting key words.

---

## “Media Analysis”

**Using the 5-layer concept**

(Image: Hard drives, USB-drives, flash memory drives ...)

---

### The 5 Layers

- **Physical Layer**
  - Media (e.g. Hardware identification: size, type, format, vendor)

- **Data Layer**
  - Info about the boot sector structure, partitioning, type of file system

- **File System Layer**
  - Specific information about files and directories

- **Metadata Layer**
  - Information extracted from file Table (e.g. Inode, Fat, MFT)

- **File Layer**
  - Analysis of information from Files (Artifact identification)
The 5-layer analysis concept of any media.

I always recommend that we should analyze any media by using the 5 layer concept.

---

The 5 Layers – main tools

- **Physical Layer**
  - Tools: fdisk, sfdisk

- **Metadata Layer**
  - Tool: ifind, ffind, istat, ils-sleuthkit, fis

- **File System Layer**
  - Tools: fsstat, jls

- **Data Layer**
  - Tools: file, testdisk, mmls, mmstat, mmcat, img_cat, img_stat

- **Physical Layer**
  - Tools: fdisk, sfdisk

---

“Physical Layer”
(Analysis of Information from media and/or image)

The first Layer – Physical Layer

This is where the Expert should gather and document information about related data storage devices, such as:

- Hard disk drives
- Removable media
- Size, vendor, type...
Defining the physical layer

The physical layer is the stage when media information is collected.

This information will be used in the record during the Chain custody process, that's it!

The next layer, the data layer, is where specific information about how the media is structured, is collected.

In this layer the expert's actions are very clear and straightforward.

The expert must identify the media partition structure or find out if it needs to be recovered.

“Data Layer”
(Analysis of information from boot sector and partitioning)

The preliminary step for this phase of the analysis happens when information is gathered from a storage device, bit by bit.

This is where the integrity of the generated images is assured through the verification of the partition information and the file system structure.

Data Layer: Useful Tools

These collect basic hard disk info:
- disk_stat
- disktype
- file
- scsiinfo

These show partition info from HD or image:
- fdisk
- sfdisk

This shows partition and slackspace info from HD or image:
- mmls
Example of tools that can be used in this layer

Data Layer: Useful Tools

- **testdisk**

  These collect hard disk or image statistic info:
  - **img_stat**
  - **mmstat**

  These allow manipulation of images and HD:
  - **mount**
  - **losetup**

The testdisk is a great tool for recovering or identifying information about partition structure and slackspace between partitions.

Example of File usage

```bash
file -s /dev/sda
/dev/sda: x86 boot sector; GRand Unified Bootloader, stage1 version 0x3, stage2 address 0x2000, stage2 segment 0x200; partition 1: ID=0x83, active, starthead 1, startsector 63, 8384512 sectors; partition 2: ID=0x8e, starthead 0, startsector 8385930, 147910455 sectors, code offset 0x48
```

Example of LSHW command use

```bash
#lshw
#
c4it0c4.4lin.com.br
  description: Desktop Computer
  product: System Product Name
  vendor: System manufacturer
  version: System Version
  serial: System Serial Number
  width: 32 bits
  capabilities: smbios-2.3 dmi-2.3 smp-1.4 smp
  configuration: boot=normal chassis=desktop cpus=2 uuid=18F67DE5-87FE-D511-A9F8-E16BAE8F0FD3
    - core
      description: Motherboard
      product: PSPE-VM
      vendor: ASUSTeK Computer Inc.
      physical id: 0
      version: Rev 1.00
      serial: MB-1234567890
```

This gets relevant information such as:
- boot Sector default
- ID Linux Partition
- ID Linux LVM Partition
This gets general information about all hardware with lshw command

Example: Vendors, pci devices, chipsets...

Data Layer

Get static info from device with DISK_STAT

disk_stat /dev/sda
Maximum Disk Sector: 156301487
Maximum User Sector: 156301487
  0 - 0 0 Empty

Data Layer

Get SCSI info from /proc/scsi/info

# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: ATA      Model: ST380013AS       Rev: 3.18
  Type:   Direct-Access                    ANSI SCSI revision: 05
Host: scsi1 Channel: 00 Id: 00 Lun: 00
  Vendor: ATA      Model: ST380013AS       Rev: 3.18
  Type:   Direct-Access                    ANSI SCSI revision: 05

Data Layer

Get info from device with SCSIINFO

scsiinfo -a /dev/sda
Scsiinfo version 1.7(eowmob)

Inquiry command
-------------------
Relative Address 0
Wide bus 32 0
Wide bus 16 0
Synchronous neg. 0

-------------------
Vendor: ATA
Product: ST380211AS
Revision level: 0.1A
Serial Number: 'SPSCVHN'
Unable to read Rigid Disk Geometry Page 04h
Data from Caching Page
Get info from Image with FDISK / SFDISK

First, it is necessary to analyze the partition structure of the image to be investigated using the following commands:

# fdisk -lu image.img
# sfdisk -luS image.img

This gets general information about partitions.

An example of how to collect information about partition structure from device with fdisk command:

Look:
Size disk
Heads
Sectors
Cylinders
Default sector size – 512 bytes

Get info from device with FDISK

fdisk -lu /dev/sda

Disk /dev/sda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders, total 156301488 sectors
Units = sectors of 1 * 512 = 512 bytes
Disk identifier: 0xcb0acb0a

Device Boot Start      End   Blocks   Id  System
/dev/sda1   *     63    8384574   4192256   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2   8385930  156296384  73955227+  8e  Linux LVM

An example of how to collect information about partition structure from device with fdisk command:

Look:
Size disk
Heads
Sectors
Cylinders
Default sector size – 512 bytes

Get info from image with FDISK

fdisk -lu /dev/sda

Disk /dev/sda: 0 MB, 0 bytes
16 heads, 63 sectors/track, 0 cylinders, total 0 sectors
Units = sectors of 1 * 512 = 512 bytes
Disk identifier: 0x00000000

Device Boot Start      End   Blocks   Id  System
/dev/sda1   *     63      72575   36256+  83  Linux
/dev/sda2    72576   2116799  1022112    5  Extended
Partition 2 has different physical/logical endings:
phys=(1023, 15, 63) logical=(2099, 15, 63)
/dev/sda5    72639   278207   102784+  83  Linux
/dev/sda6   278271   410255    65992+  82  Linux swap / Solaris
/dev/sda7   410319   513071    51376+  83  Linux
/dev/sda8   513135  2116799   801832+  83  Linux
An example of how to collect information about partition structure with fdisk from image.

Similar to fdisk, an example of how to collect information about device partition structure with sfdisk.

Now

An example of the MMLS command, output of hard disk image
Another example of MMLS from image

Look again; slackspace partition information in red

Example of DISKTYPE command use

```
# disktype /dev/sda
--- /dev/sda
Block device, size 74.53 GiB (80026361856 bytes)
GRUB boot loader, compat version 3.2, boot drive 0xff
DOS/MBR partition map
Partition 1: 3.998 GiB (4292870144 bytes, 8384512 sectors from 63, bootable)
  Type 0x83 (Linux)
  Ext3 filesystem
  UUID 0A40FE81-CD61-452B-91F5-0FDA1F2EAB50 (DCE, v4)
  Volume size 3.998 GiB (4292870144 bytes, 1048064 blocks of 4 KiB)
Partition 2: 70.53 GiB (75730152960 bytes, 147910455 sectors from 8385930)
  Type 0x8E (Linux LVM)
  Linux LVM2 volume, version 001
  LABELONE label at sector 1
  PV UUID 0BV3m3-qoZM-Zgrb-gw38-Mdbr-QcMX-x32Q6U
  Volume size 70.53 GiB (75730152960 bytes)
  Meta-data version 1
```

Another way to collect device storage information using disktype command,
Example information: "boot loader type, partition type, partition size, LVM information, volume size"

DEMO

"Filesystem Layer" (For use in file system structure analysis)

File System Layer
In this phase you need to collect information about the file system, example:
- Type
- size info.
- Date of last access
- Date of last write
File System Layer: Useful Tools

Common tools to collect info from the File system. This gets journal info from image, (e.g. statistics about partition).

- **Fsstat**
  - This shows general info from journaling file system
  - **jcat**
  - This shows journaling info from structure of file system
  - **jls**

Examples of interesting tools for the expert to use in this phase.
- Fsstat
- Jcat
- Jls

**File System Layer**

Example of **FSSTAT** command use

```
# fsstat image.img
FILE SYSTEM INFORMATION
----------------------------
File System Type: Ext3
Volume Name: /
Volume ID: ef3c387a7bc4ac9fdb1140dcec080dae
Last Checked at: Tue Mar 27 05:53:49 2007
Unmounted properly
Last mounted on:
Source: OS: Linux
Dynamic Structure: Journal,
Compat Features: Filetype, Needs Recovery,
Read Only Compat Features: Sparse Super,
```

**File System Layer**

Example of **JCAT** command use (e.g. 3001 inode)

```
# jcat -f ext tambaquicorp.img 3001
```

```
.?
..??
km3xsadan.sh>
sadan.sh.1?
```
**Example of JLS command use**

```bash
# jls -f ext tambaquitcorp.img | tail -n 10
4086: Allocated FS Block 164013
4087: Allocated FS Block 163957
4088: Allocated FS Block 163962
4089: Allocated FS Block 105
4090: Allocated FS Block 131115
4091: Allocated FS Block 163860
4092: Allocated FS Block 65572
4093: Allocated FS Block 65576
4094: Allocated FS Block 65584
4095: Allocated FS Block 65589
```

---

**“Metadata Layer”**

(Analysis Inode Table information)

---

**Metadata Layer**

Once we have accessed the file system, the search for previously accessed files - or even files already input into the system - can be initiated, allowing us to search for evidence related to the incident.

The metadata analysis information is an extremely important step in the search for evidence and other actions in the fifth layer (File Layer).
**Useful Metadata Tools**

These show Inode structure info
- istat (static info)
- ils
- ifind
- This collects content of a specific Inode
- icat
  This collects mactime info of all files in the Inode table and allows us to create the timeline.
- fls
- mactime

---

**The all important timeline**

This is a large report with all file info and its mactime:

The timeline is created based on **MACtime** (Modified, Accessed, Created|Changed)
Info about when:
- the Operating system (O.S.) was installed.
- Changes and updates were made
- the O.S. Was last used
- and many other details related to the manipulated filesystem’s files.

---

**Sleuthkit Timeline creation**

Example of how to create a hard disk image timeline

```bash
# fls -alrpm / image.img | tee body
# mactime -b body
```

How to create a specific period timeline

```bash
# fls -alrpm / image.img | mactime -z GMT-3 01/01/2000 01/01/2002 | tee timeline.txt
```

---
Sleuthkit Timeline creation

How to create a mounted image timeline:

```
# mount /media/imagem -o loop,noexec,nodev,noatime,ro
# fls -r /media/imagem | mactime -z GMT-3 01/01/1970 09/08/2007 | tee timeline.txt
```

This is an example of sleuthkit timeline creation of a mounted image.

---

Sleuthkit Timeline creation

How to create a mounted image timeline of a specific interval:

```
# fls -r image.img | mactime -z 01/01/2006 09/08/2007 | tee timeline.txt
```

```
This is an example of sleuthkit timeline creation of a specific interval.
```

---

Metadata Searching

Exemplifying information collection from an allocated area.

And following, how to create a file with strings from allocated info:

```
# dls -a -f image.img > image.img.dls
# strings -e image.img.dls > image.img.dls.alocadas.strings
# less image.img.dls.alocadas.strings
```

---

DEMO
Here we have an example of using sleuthkit to extract information about allocated files.

```
# dis -A -f ext image.img > image.img.dls
# strings -a image.img.dls > image.img.dls.naoalocadas.strings
# less image.img.dls.naoalocadas.strings
```

This is an example of using sleuthkit to extract information about unallocated files, with the dis command.

Metadata Searching
Exemplifying information collection from an unallocated area. And following, how to create a file with strings from unallocated info:

```
# dis -A -f ext image.img > image.img.dls
# strings -a image.img.dls > image.img.dls.naoalocadas.strings
# less image.img.dls.naoalocadas.strings
```

"File Layer" (Analysis of file information and identification of possible artifacts)

Data Blocks: useful tools
Shows statistical info from data blocks
- dstat
Enables us to list info from allocated, unallocated and slackspace areas
- dis
- dcat
Manipulate info from a specific data block
- dcalc
Example of tools:
- Dstat
- Dls
- Dcat
- Dcalc

Tools for File Layer analysis
Enables one to consult file and directory information from an image, using metadata.
fls

Similar to fls but using the specific Inode address.
ffind

Enables one to sort the files according to their type.
sorter

Enable one creates and searches and indexed database hash
hfind

Enables searches for hex and signature at any specified offset
sigfind

More Examples of tools

Look at commands:
fls
ffind
Sorter
hfind
sigfind

Image Mounting

It's recommended that disk forensic image analysis be a process executed with caution, beginning with a media access preparation known as “mounting”.

The image mounting of the partition with the means of analysis must be accessed as a read-only filesystem, without device file and executable file support.
Partition mounting is a very relevant action in this phase. The media mounting process is simple but the expert can’t forget the three main steps to execute it:
- disable support to execute files
- disable support to device files (only for system like Unix)
- read only always

Example of image mounting of a single partition

```bash
# mount /pericia/imagem.img /img/ -t ext3 -o loop,ro,noatime,nodev,noexec

# mount | tail -1
/pericia/imagem.img on /img/ type ext3
(rw,noexec,nodev,loop=/dev/loop1)
```

Example of image mounting of multiple partitions

When dealing with this specific subject, it’s necessary to analyze all hard disk images using losetup command.

```bash
# losetup /dev/loop0 /imagem_hd.img
```

Another important detail is, if there is an image with multiple partitions, it is necessary to use the losetup tool to mount each image partition.
Example of image mounting of a partition with losetup

In a given scenario, where the mounting of a second listed partition is required, let’s suppose that the initial sector of the partition is 73. Considering this case, this value must be multiplied by 512 to calculate the offset value.

Expr: 73 * 512

The result determining the offset value is 37376

Mouting a partition from the full disk image

Before the full disk image analysis, it’s necessary to understand the status of the image partitioning structure:

# sfdisk -lu /dev/HD_coleta.img
read failed: Inappropriate ioctl for device
Disk HD_coleta.img: cannot get geometry
Disk HD_coleta.img: 171 cylinders, 255 heads, 63 sectors/track
Warning: extended partition does not start at a cylinder boundary.
DOS and Linux will interpret the contents differently.
Warning: The partition table looks like it was made for C/H/S = */16/63 (instead of 171/255/63).
For this listing I’ll assume that geometry.
Units = sectors of 512 bytes, counting from 0

Gathered info about all partitions

<table>
<thead>
<tr>
<th>Device</th>
<th>Boot</th>
<th>Start</th>
<th>End</th>
<th>#sectors</th>
<th>Id</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD.img1</td>
<td>*</td>
<td>63</td>
<td>72575</td>
<td>72513</td>
<td>83</td>
<td>Linux</td>
</tr>
<tr>
<td>HD.img2</td>
<td></td>
<td>72576</td>
<td>2116799</td>
<td>2044224</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Extended</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HD.img3</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Empty</td>
<td></td>
</tr>
<tr>
<td>HD.img4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Empty</td>
<td></td>
</tr>
<tr>
<td>HD.img5</td>
<td>72619</td>
<td>278207</td>
<td>205569</td>
<td>83</td>
<td>Linux</td>
<td></td>
</tr>
<tr>
<td>HD.img6</td>
<td>278271</td>
<td>410255</td>
<td>131985</td>
<td>82</td>
<td>Linux</td>
<td></td>
</tr>
<tr>
<td>swap / Solaris</td>
<td>410319</td>
<td>513071</td>
<td>102753</td>
<td>83</td>
<td>Linux</td>
<td></td>
</tr>
<tr>
<td>HD.img8</td>
<td>513075</td>
<td>2116799</td>
<td>1603665</td>
<td>83</td>
<td>Linux</td>
<td></td>
</tr>
</tbody>
</table>
Preparation for mounting of partition with losetup

```
# losetup -a
# expr 410319 * 512
210083328
# losetup -o 210083328 /dev/loop2 HD_coleta.img
```

Finally, calculate the offset and use the losetup to link to the special loop device.

mounting of partition with losetup

```
# losetup -a
/dev/loop2: (/dev/fd01:131073 (/home/c4/DIGITAL_FORENSIC/forensic_duplicate*), offset 210083328
# mount -t ext2 /dev/loop2 /media/loop0p2 -o loop,loopexec,nodev
# cd /media/loop0p2
# ls
arpwatch cache db ftp lib local lock lost+found mail nis opt preserve run spool tmp www yp
```

Once the loop device is defined, the mounting process can be executed.

This shows mounted partition info

- `df`
- Filesystem 1k-blocks Used Available Use% Mounted on
- `/dev/swap1` 41294860 4924120 34273056 13% /
- `/dev/mapper/vg_zhekeki-LV_home`
- 146166336 7445736 131295784 6% /home
- `/dev/loop2 /media/loop0p2`
- `tmpfs` 1026832 1020 1025812 1% /dev/shm
Mounting the image

But for the whole hard disk image analysis, it is necessary to use the losetup command:

```
# losetup /dev/loop0 /imagem_hd.img
```

Arranging files by type

An important action is to list all files in the analyzed media, arranging them according to format.

For this task, SORTER command is the recommended tool.

Using sorter and losetup commands together

Here is an example of the use of the sorter command straight from a device prepared with the losetup command.

```
# losetup /dev/loop0 image.img
# sorter -f ext -l /dev/loop0
```
**Uses of find command**

Search for files with SUID and SGID permission that can be used in Malware, such as backdoors:

```bash
# find / -perm -04000
# find / -perm -02000
# find / -type f (-perm -04000 -o -perm -02000) -exec ls -l {} ;
```

The next step is to search for files with specific characteristics such as:

- With special permission

The following slides are some examples of the "find" command that show:

**Search for artifacts with FIND**

Search for files and directories that have a name using a blank space:

```bash
# find / -name "*[ ]*"
```

The first one:
The search for files and directories with "blank" in your name.

**Search for artifacts with FIND**

Search for hidden files and directories like Unix, that is, files that begin with ".", which in a system such as Unix characterizes a file or directory as hidden.

This is a very common procedure used to find info on possible tools used by an intruder:

```bash
# find / -type f (-name ".??*" -o -name ".[^.]*") -exec ls -l {} ;
```

The next one:
The search for files and directories with "dot" and "blank space" in your name.
Search for artifacts with FIND

Search for files without owner or specified group, that can be installed in the system unconventionally:

```
# find /img/ -nouser
# find /img/ -nogroup
# find /img/ -type f \(-nouser -o -nogroup \) -exec ls -ldg {} \;
```

This example the search for files and directories without owner (user) and group.

Search for artifacts with FIND

Many intruders try to hide info in system directories that are for specified data and are not constantly accessed. An example would be directories such as /dev and /lib:

```
# find /dev/ -not -type c -not -type b ls -l
```

another one
Now, find the file not character and block files inside /dev directory

Search for artifacts with FIND

Searching for files that are access or metadata time modified after the time of a specified file, is another kind of search that should be performed since it can enable the identification of other potential artifacts:

```
# find / -anewer /etc/shadow ls -lha
# find / -cnewer /etc/shadow ls -lha
```

Next one, collect information about files from the same "time of access - answer" or "change access - chewer" of another file.
Searching for artifacts with FIND

Searching for files whose access time is within a determined time frame. This kind of search is also useful for artifact identification, in which case searching for atime and mtime is interesting:

```
# find /img/ -atime 3 ls -lha
# find /img/ -ctime 3 ls -lha
# find /img/ -mtime 3 ls -lha
# find /img/ -mtime 3 -or -atime 3 ls -lha
```

And finally, one more example using access, change and modify time as a reference.

Searching for Malware

There are two interesting tools used for searching the well known “rootkits” in the system “chkrootkit” and “rkhunter” which identify signs that the machine has been infected.

```
# chkrootkit -r /img/
```

The task of identifying malware is also a procedure that should always be done. For this task the expert can use two types of tools.

- Rootkit scanners
- Anti virus

```
# rkhunter --check --sk --rwo --rootdir img/ --createlogfile rkhunter_forensic.log
```

this is an example of rkhunter
Searching for Malware

Searching for Malware info with "clamav" command:

```
# clamscan -i -r -d /result img/
```

And the last one, an example with anti-virus clamscan

**Slackspace Evidence**

Searching for evidence in slackspaces

---

We can't forget the importance of slackspace recovery, because sometimes we can find evidence. For example:
- parts of an email
- parts of a file

Searching Slackspace

Slack space in file (data blocks) is a very important source of evidence in computer forensic investigation/

It is recommended that an exclusive extraction be done, keeping in mind that any computational evidence can be both very small AND very significant (such as the 4 bytes of an IP address).
Slackspace recovery is a simple procedure which already happens when the expert creates the string map of hard disk image.

```
# dls -s image.img | slackspace.dls
# strings -a slackspace.dls > slackspace.dls.strings
```

This is an example of how to collect slackspace information with dls command.

The File Carving process is imperative, but the expert needs to distinguish the relevant files from the irrelevant ones, because file carving delivers a massive amount of files.

**Investigating Slackspace**

These allow us to get information about slackspace from an image:

```
# dls -s image.img | slackspace.dls
# strings -a slackspace.dls > slackspace.dls.strings
```

**“File Carving Techniques”**

Analysis of unallocated areas that may contain relevant artifacts.

**Recovery**

File recovery is a necessary activity in practically every Post Mortem. However, this task demands specific tools.

Luckily, an Expert has several options when it comes to FLOSS tools.
File recovery is a necessary activity in practically every Post Mortem. However, this task demands specific tools.

Another relevant point is the fact that some file systems not only perform the unlinking of the metadata and the data, but also overwrite the metadata with zeroes. Example: EXT3

Useful tools for recovery

- **Magicrescue**: together with DLS, this permits the recovery of the files. It foremost this recovers files from their headers and footers.
- **ddrescue**: this recovers files from the image of any medium, but is a mode hard. It’s necessary identify file offset address.

Examples of recovery tools

Look:
- Magicrescue
- Foremost
- Ddrescue

File recovery using classic procedure

Attempting to recover a file from an image:

a) Identify the addresses using metadata of unallocated files

```
# fls -t ext.img > list.image.txt
```

b) Retrieve content from the list (unallocated files)

```
# cat list.image.txt
```

c) Recover it by using the ICAT command with specific content file by inode (e.g. 4157)

```
# icat image.img 4157 > file.ppt
```
This is a step by step process, where you first get a list of all unallocated files, and second you choose the specific file (by inode) and finally you recover your content.

Recovery with Foremost

One way to recover files is by using FOREMOST, which automatically performs a complete analysis of the file system.

```
# foremost -c foremost.conf -i image.img -o /recovery -T
```

However, the expert should use automated tools to do this, such as foremost.

Recovery with Foremost

Another way to use FOREMOST is to perform a search for types of file. Examples for images (e.g. jpg, gif, png), for PDF:

```
# foremost -c foremost.conf -t jpeg,png,gif,pdf -v -i image.img -o /recovery -T
```

This is an example with foremost.
So, there are many tools for the Post Mortem Process. As well as using automated tools, we have the “5Layer Approach” to allow us to do a more detailed analysis, also when the available tools are unable to help, and we need to do a “hands on” analysis.

The Whole 5-Layer Process

Start:
Copy bit by bit from media to chain of custody

Data Layer:
Physical Layer
Identify Information About Partition Structure And ID Partition

File System Layer:
Identify Information About File System In Use

Metadata Layer:
Identify Information About Files Allocated And File Unallocated

File Layer:
Identification Of Artifacts Recovery Of File Identification Malware

String Extracts & Analysis with Regex

Artifacts Analysis

Report

File Carving

File Layer (3rd)

Metadata Layer (4rd)

File System Layer (3rd)

Data Layer (2nd)

Physical Layer
Media/Images Info (1st)

So, there are many FLOSS tools CLIS (Command Line On Steroids) and also GUI Tools (example: Autopsy, Pyflag, PTK) for the Post Mortem Process, and by combining the 5 Layer Concept with String Extraction it is possible to analyze everything related to an Incident.

Another fact is that the Linux OS is the better choice for Computing Forensics, because it supports many filesystems and you can customize your Forensic Box.

Remember!!! - Incidents will happen and you need to be ready we sysadmin need to learn from our mistakes so that they are not repeated.

The force is with us

CONCEPTS

Sandro Melo – sandro.melo@bandtec.com.br - Brasil – 197
ANY QUESTIONS?

Some beer?

Any Questions

Who’s paying for the beer?