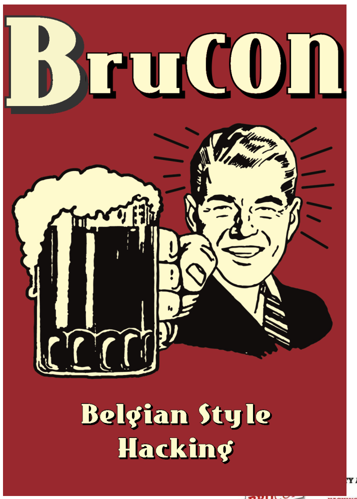
Kudo Daido Juku allows the fighter to adapt to each new situation using the Budo spirit as their guide



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).

My email contact

in the next few slides of this presentation you will find my brief resume.

That you can take a look at later

About me

About Sandro Melo - aka CARIOCA - Currently I work at Bandtec College, and also with Advanced Training, Pentest, Response to Security Incidents and Computer Forensic and student/candidate in Doctor Program in TIDD/PUC-SP. I was born in the beautiful city Rio de Janeiro, Brazil. I moved to Sao Paulo where I began my professional career in System Security. Since 1996 I have worked mainly with Linux/ FreeBSD and FLOSS (Free/libre and Open Source Software), Network Administrator, I am often a guest professor at many universities all over Brazil. Project Fedora Linux Ambassador, LPI and BSDA PROCTOR.

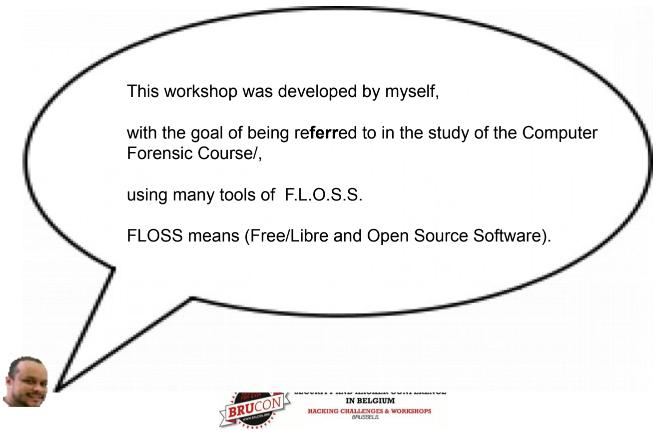
I take great pride in everything I do, especially with my work in Forensics. I have years of hands-on experience with many of the core technologies and have written many books and articles on security and forensics. When not working or writing, I can be found experimenting with the latest Open Source solutions, installing 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.

"Ik ben zeer blij hier in BruCON / J'ai très heureux ici dans BruCON"

Hi guys, good morning.
I'm Sandro Melo from Sao Paulo Brazil
If you have any doubts, I'm the one on the right
Being from Brazil, English isn't my first language,
so I apologize now for any mistakes I make.
If there is something you don't understand, a transcript of my talk is **available.**
Let's begin on the next slide

" HANDS ON KUDO - POST MORTEM FORENSIC ANALYSIS with specific Forensic FLOSS TOOLS – 2.0"

CONCEPTS



BandTec

Introduction

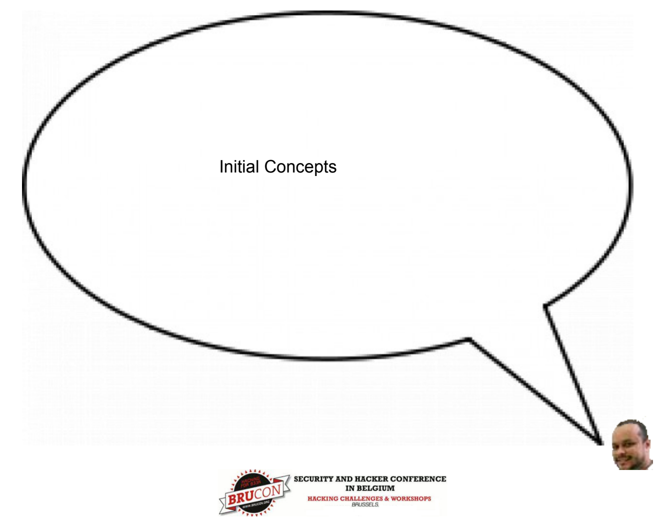
In the past, servers configured their risks but these risks were physically dimensioned, corresponding to the limits of the LAN of the corporation or institution. The Internet has radically changed this scenario.

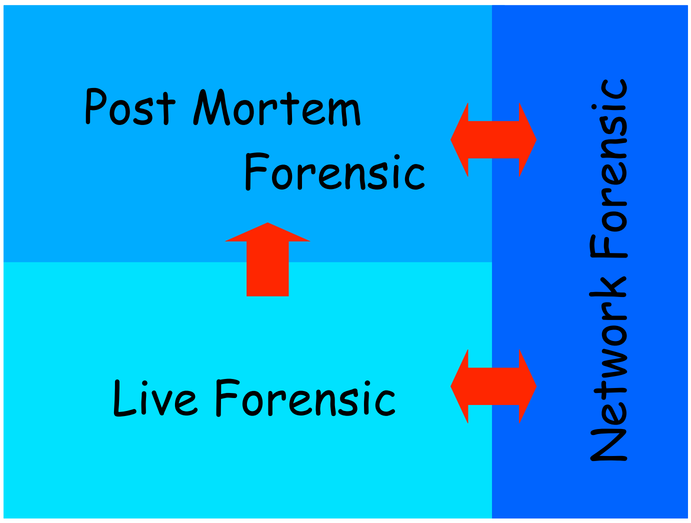
It is more secure than a system with Firewall or other security devices, there will always be the possibility of human error or hitherto unknown failure in the operating system or applications, whether proprietary or FLOSS system. Given this degree of risk, at first intangible, the threat of an invasion 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 identity 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.

SECURITY AND HACKER CONFERENCE IN BELGIUM
HACKING CHALLENGES & WORKSHOPS





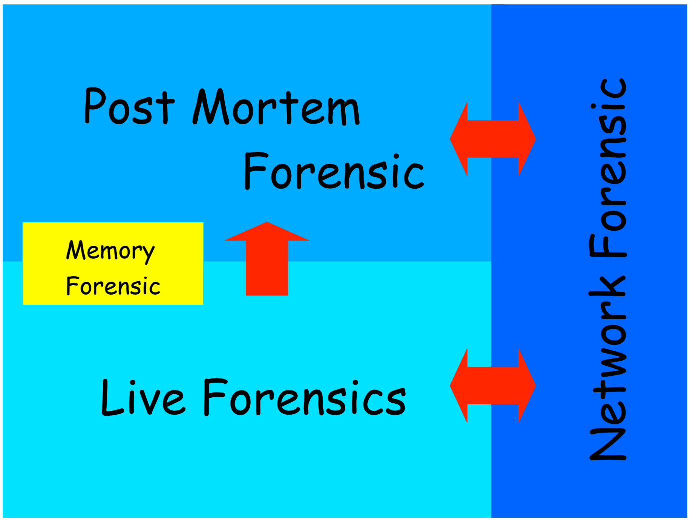
We can divide the process into 3 phases.

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.

SECURITY AND HACKER CONFERENCE IN BELGIUM



The 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 Forensic.

SECURITY AND HACKER CONFERENCE IN BELGIUM

Image for Post Mortem

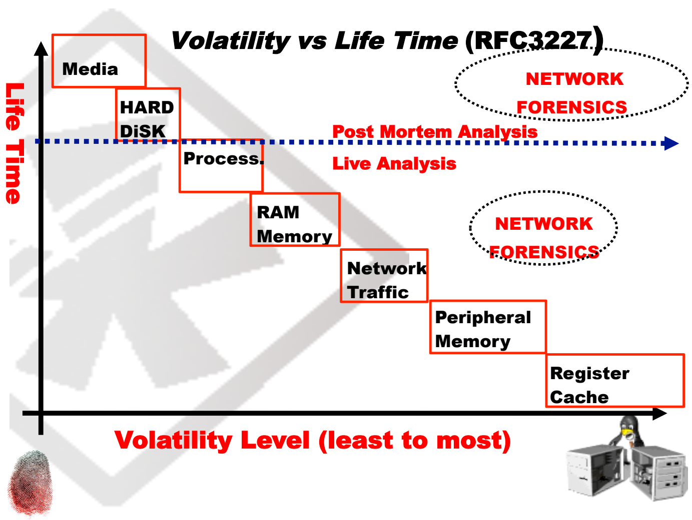
Practically the whole Post Mortem begins when the file image is created. The image can be created in specific formats such as:

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

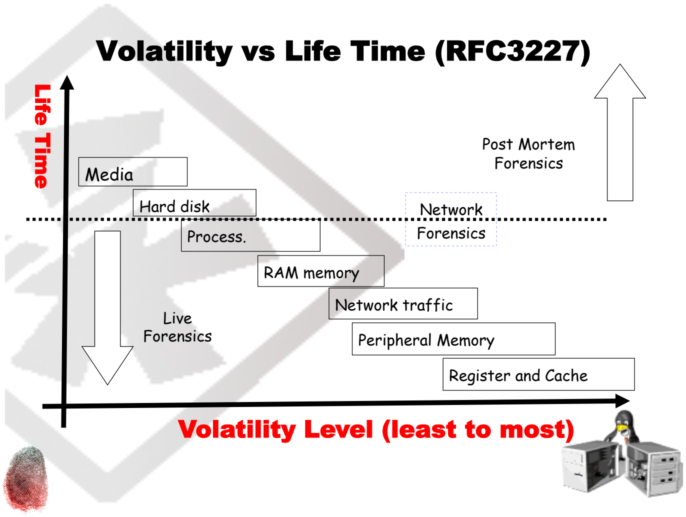
There are specific formats to create a image forensic

SECURITY AND HACKER CONFERENCE IN BELGIUM



The Live Forensic represents data collection while the device is turned on.

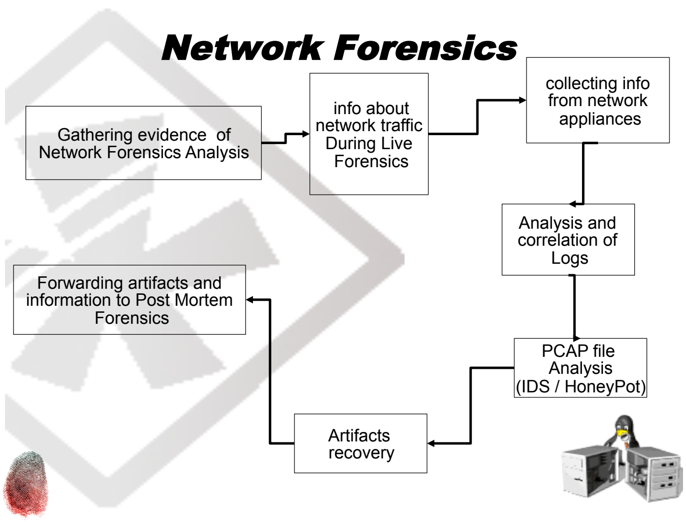
This information shows a certain degree of **volatility** that must be considered during the process.



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?

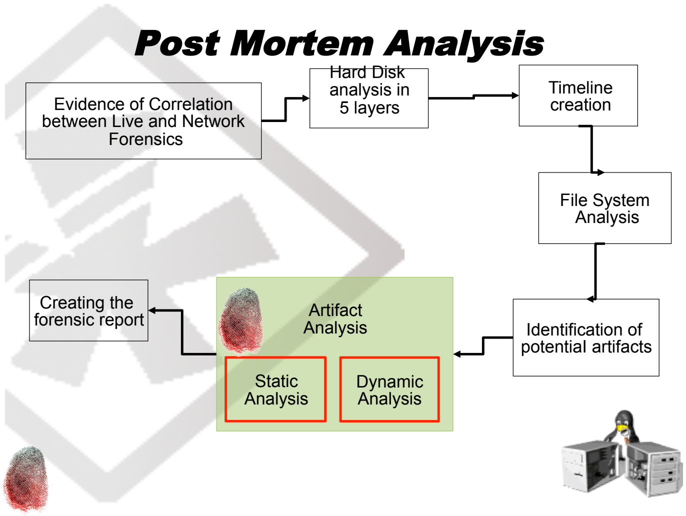
So, if an incident occurs in a network without security assets, such as firewall, proxy, IDS, IPS, logs server, you will need to collect more information because if we only have a gateway, little or no relevant information can be collected in this phase.

(Brushing bits, data mining, seeking for Evidence and Artifacts)

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 **ex**citing. Why?

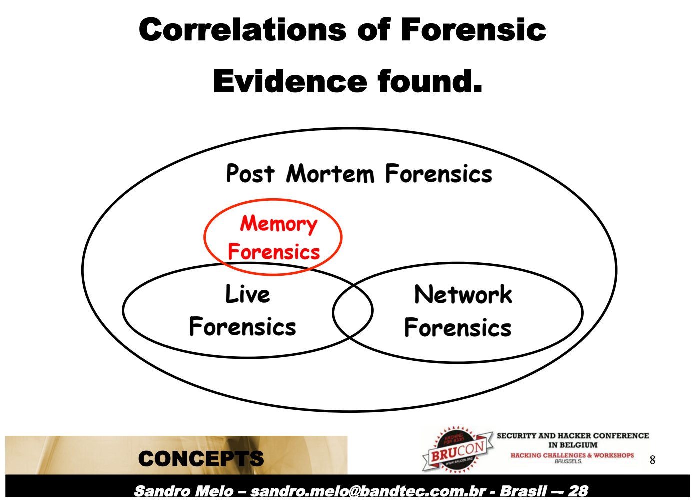


Because most of the time, all the information coming together, creates a clearer picture.
The proposal of this workshop is to suggest that ana**ly**sis of the media image,

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.

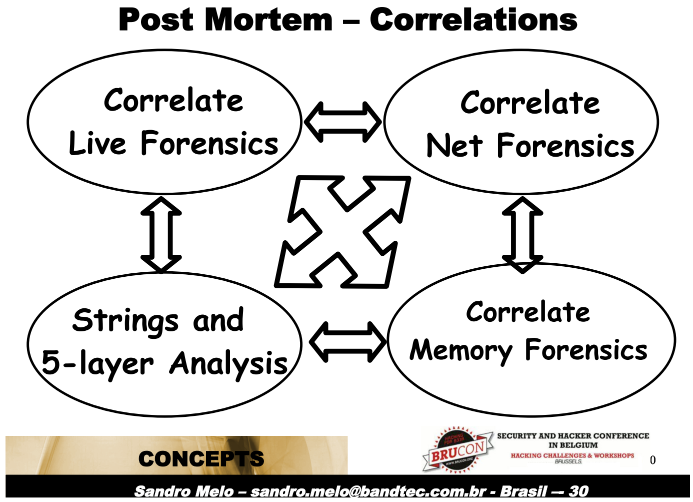


Correlating the information collected during Live, Network, Memory with Post Mortem Analysis is the most important action, so I recommend you do this as much as possible.

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.



The information link in each phase is the most important point during all analyses of the incident. However, not having Live, Network and Memory Forensic data, doesn't make it impossible to execute Post Mortem analysis and doesn't mean that it won't be productive, because the hard disk is a type of media that contains a world of information in which potential evidence can be identified.

Correlation is a great tool that the expert has to use during the whole Post Mortem process. Possible clues can be found in extracted strings and other forensics phases, And must also be correlated with clues found in each layer of specific media images, to culminate in relevant evidence.

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.

"When we know the enemy, we need not fear the outcome of 100 battles" - Chinese proverb.

When the "bad guys" are known it is easier to identify the evidence. It is good to have in mind that the Modus Operandi of the bad guys is the main reason for forensic studies, based on honeypot. Which is essential nowadays.

We need to learn about "Modus Operandi: of "Bad Guys"

Initial System Analysis

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

The better the technical knowledge of the bad guys, the more difficult it is to identify relevant evidence of their actions.

In the same way, if the bad guys have little knowledge, the chances of identifying the evidence is better.

Therefore the modus operandi is very important and must be updated constantly.

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 hhy 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 identific**ation** 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 **in**teresting

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

Strings vs 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

Extracting strings through key words

A practical way to do this is through the generation of a file with key words and usual expressions, aiming to automate the search.

```
# 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 ...)

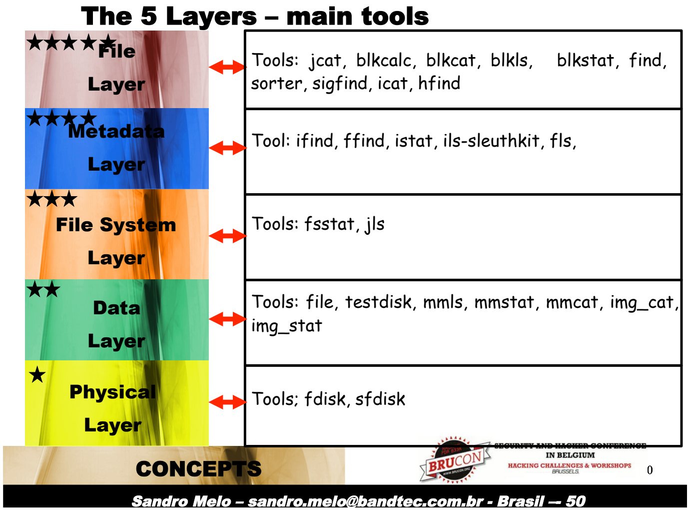
The media analysis process

The 5 Layers

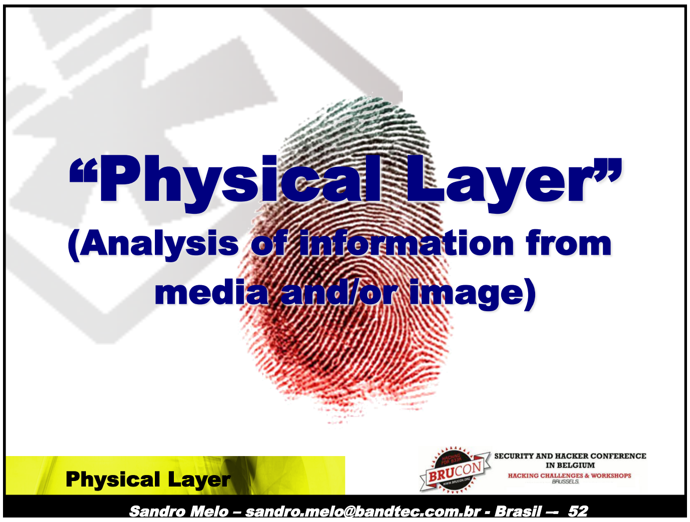
★★★★★ File Layer	Analysis of information from Files (Artifact identification)
★★★★ Metadata Layer	Information extracted from file Table (e. g. Inode, Fat, MFT)
★★★ File System Layer	Specific information about files and directories
★★ Data Layer	Info about the boot sector structure, partitioning, type of file system
★ Physical Layer	Media (e.g. Hardware identification: size, type, format, vendor)

The 5-layer analysis concept of any media.

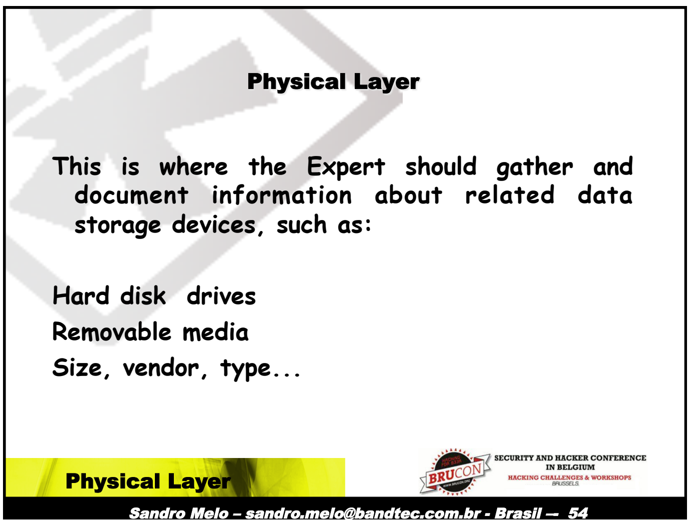
I always recommend that we should analyze any media by using the 5 layer concept



Look at these main tools that can be used in each layer



The first Layer – Physical Layer



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!

"Data Layer"

(Analysis of information from boot sector and partitioning)

The next layer, the data layer, is where specific information

about how the media is structured, is collected.

Data Layer

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.

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: 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

This allows us to see partition info and if necessary to recover partition structure:

- testdisk

These collect hard disk or image statistic info:

- img_stat
- mmstat

These allow manipulation of images and HD

- mount
- losetup

Data Layer

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

Example of File usage

```
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
```

Data Layer

This gets relevant information such as :

- boot Sector default
- ID Linux Partition
- ID Linux LVM Partition

Example of LSHW command use

```
#lshw
c4ri0c4.4nix.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-B7FE-
D511-A9F8-E16BAE8F0FD3
  *-core
     description: Motherboard
     product: P5PE-VM
     vendor: ASUSTeK Computer Inc.
     physical id: 0
     version: Rev 1.00
     serial: MB-1234567890
```

Data Layer

This gets general information about all hardware with lshw command

Example: Vendors, pci devices, chipsets…

Get static info from device with DISK_STAT

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

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

Data Layer

Simple statistics information about sectors

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

This gets general information about scsi devices

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:                   3.AA

Serial Number '                   5PS0GVN0'
Unable to read Rigid Disk Geometry Page 04h
Data from Caching Page
```

Data Layer

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
```

Data Layer

SECURITY AND HACKER CONFERENCE IN BELGIUM
HACKING CHALLENGES & WORKSHOPS

This gets general information about partitions

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
```

Data Layer

SECURITY AND HACKER CONFERENCE IN BELGIUM
HACKING CHALLENGES & WORKSHOPS

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 HD_coleta.img
read failed: Inappropriate ioctl for device
You must set cylinders.
You can do this from the extra functions menu.
Disk HD_coleta.img: 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
HD_coleta.img1   *        63    72575     36256+   83  Linux
HD_coleta.img2         72576  2116799    1022112    5  Extended
Partition 2 has different physical/logical endings:
     phys=(1023, 15, 63) logical=(2099, 15, 63)
HD_coleta.img5         72639   278207    102784+   83  Linux
HD_coleta.img6        278271   410255     65992+   82  Linux swap / Solaris
HD_coleta.img7        410319   513071     51376+   83  Linux
HD_coleta.img8        513135  2116799    801832+   83  Linux
```

Data Layer

SECURITY AND HACKER CONFERENCE IN BELGIUM
HACKING CHALLENGES & WORKSHOPS

An example of how to collect information about partition structure with fdisk from image

Get info from device with SFDISK

```
# sfdisk -luS /dev/sda

Disk /dev/sda: 9729 cylinders, 255 heads, 63 sectors/track
Units = sectors of 512 bytes, counting from 0

   Device Boot    Start       End   #sectors  Id  System
/dev/sda1   *        63   8384574    8384512  83  Linux
/dev/sda2       8385930 156296384  147910455  8e  Linux LVM
/dev/sda3             0         -          0   0  Empty
/dev/sda4             0         -          0   0  Empty
```

Data Layer

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

Get info from device with MMLS

```
# mmls /dev/sda
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

    Slot      Start        End          Length       Description
00:  Meta     0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----    0000000000   0000000062   0000000063   Unallocated
02:  00:00    0000000063   0008384574   0008384512   Linux (0x83)
03:  -----    0008384575   0008385929   0000001355   Unallocated
04:  00:01    0008385930   0156296384   0147910455   Linux Logical Volume
Manager (0x8e)
05:  -----    0156296385   0156301487   0000005103   Unallocated
```

Data Layer

Now

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

Here, You can see slackspace partition information in red

Get info from image with MMLS

```
mmls HD_coleta.img
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors
    Slot      Start        End          Length       Description
00:  Meta     0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----    0000000000   0000000062   0000000063   Unallocated
02:  00:00    0000000063   0000072575   0000072513   Linux (0x83)
03:  Meta     0000072576   0002116799   0002044224   DOS Extended (0x05)
04:  Meta     0000072576   0000072576   0000000001   Extended Table (#1)
05:  -----    0000072576   0000072638   0000000063   Unallocated
06:  01:00    0000072639   0000278207   0000205569   Linux (0x83)
07:  01:01    0000278208   0000410255   0000132048   DOS Extended (0x05)
08:  Meta     0000278208   0000278208   0000000001   Extended Table (#2)
09:  02:00    0000278271   0000410255   0000131985   Linux Swap / Solaris x86 (0x82)
10:  02:01    0000410256   0000513071   0000102816   DOS Extended (0x05)
11:  Meta     0000410256   0000410256   0000000001   Extended Table (#3)
12:  03:00    0000410319   0000513071   0000102753   Linux (0x83)
13:  03:01    0000513072   0002116799   0001603728   DOS Extended (0x05)
14:  Meta     0000513072   0000513072   0000000001   Extended Table (#4)
15:  04:00    0000513135   0002116799   0001603665   Linux (0x83)
16:  -----    0002116800   0002748977   0000632178   Unallocated
```

Data Layer

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 file system
    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
```

Data Layer

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

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 info 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

Example of FSSTAT command use

```
# fsstat image.img
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: Ext3
Volume Name: /
Volume ID: ef3c387a7bc4ac9fdb1140dcec080dae
Last Written at: Wed Mar 28 11:37:26 2007
Last Checked at: Tue Mar 27 05:53:49 2007
Last Mounted at: Wed Mar 28 11:37:26 2007
Unmounted properly
Last mounted on:
Source OS: Linux
Dynamic Structure
Compat Features: Journal,
InCompat Features: Filetype, Needs Recovery,
Read Only Compat Features: Sparse Super,
```


Look at this example of the FSSTAT

Look:
Type: Ext3
Last Written
Last Checked
Last Mounted
OS: Linux
Journal

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

```
# jcat -f ext tambaquicorp.img 3001
=
 .??
   ..??
     km3xsadan.sh>
sadan.sh.1?

-----
```


Take a look at this example of JCAT

Show the content information of inode number 3001

Example of JLS command use

```
# jls -f ext tambaquicorp.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
```


"Have a look at this - This is an example of JLS

"Metadata Layer"

(Analysis Inode Table information)

Next layer

Metadata Layer

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).

When the expert has metadata, he can get information about the times of all files, with this the expert can create the Timeline, and can also identify all unallocated files.

All of this is very important to identify the evidence..

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

These are useful metadata tools

Look commands:
Istat
Ils
Ifind
fls
mactime

The all important timeline

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

The timeline is created based on **MAC**time (**M**odified, **A**ccessed, **C**reated|**C**hanged)

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.

Creation of the timeline.

The timeline is created based on MACtime - Modified, Accessed, Created/Changed times

Timeline is a large report about all file information, mainly mactime

Sleuthkit Timeline creation

Example of how to create a hard disk image timeline

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

How to create a specific period timeline

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


This is an example of sleuthkit timeline creation,

Where you first use the fls command and then you use the mactime command.

Sleuthkit Timeline creation

How to create a mounted image timeline

```
# mount imagem /media/imagem -o loop,noexec,nodev,noatime,ro

# fls -alrpm /media/imagem /dev/loop0 | 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 -alrpm image.img | mactime -z GMT-3 01/01/2006 09/08/2007 | tee timeline.txt
```


“in front of us is an example of sleuthkit timeline creation of a specific interval

DEMO

Metadata Searching

Exemplifying information collection from an allocated area.

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

```
# dls -a -f ext image.img > image.img.dls

# strings -a image.img.dls > image.img.dls.alocadas.strings

# less image.img.dls.alocadas.strings
```

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

Metadata Searching

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

```
# dls -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 dls command.

"File Layer"

(Analysis of file information and identification of possible artifacts)

Now, let's talk about the final layer -The Final layer is "File Layer" Firstly, I need to say that this process is very important and very long, because it's necessary to analyze many types of files and correlate them with Incident information.

In this layer the expert has a lot of hard work to do, because this is where he identifies the strongest evidence, such as malware files (e.g. backdoor, exploit, rootkits, trojans), manipulated files, log files etc.

Data Blocks: useful tools

Shows statistical info from data blocks
- dstat

Enables us to list info from allocated, unallocated and slackspace areas
- dls
- 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"

Image Mounting

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

DEMO

Example of image mounting of a single partition

```
# 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)
```


This is an example of simple mounted

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.

```
# 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**

File Layer

When you have an image with multiple partitions, you need to calculate the offset address,

it's necessary to use the losetup sector partition start number and multiply by 512 bytes (default size)

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 -luS 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
```

File Layer

First of all, get the media information, (e. g. Size, format, cylinder, etc)

Gathered info about all partitions

```
Device Boot    Start       End   #sectors  Id  System
HD.img1   *       63     72575      72513  83  Linux
HD.img2        72576   2116799    2044224   5
Extended
HD.img3            0         -          0   0  Empty
HD.img4            0         -          0   0  Empty
HD.img5        72639    278207     205569  83  Linux
HD.img6       278271    410255     131985  82  Linux
swap / Solaris
HD.img7       410319    513071     102753  83  Linux
HD.img8       513135   2116799    1603665  83  Linux
```

File Layer

Then identify the sector partition start number

it is necessary to use the losetup command to mount an image with multiple partitions

Preparation for mounting of partition with losetup

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

File Layer

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

mounting of partition with loseup

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

File Layer

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/sda2 41294860 4924120 34273056 13% /
- /dev/mapper/vg_ichegeki-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

.this is an example of the link loop device with losetup

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.

Sorting files is another relevant procedure during the media analysis,

because this can be a way of identifying evidence at any given moment

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

This is an example of sort command usage

Uses of find command

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

```
# find /img –perm -04000
# find /img –perm -02000
# find /img/ -type f \(-perm -04000 -o -perm -02000 \) -exec ls -lg {} \;
```


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:

```
# find /img/ -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:

```
# find /img/ -type f \( -name '.??*' -o -name '.[^.]' \) -exec ls -lg {} \;
```


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 {} \;
```

File Layer

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 diretories such as /dev and /lib:

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

File Layer

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

File Layer

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

this is an example of chkrootkit

Searching for Malware

To search for Malware info with the command rkhunter:

 # 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:

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


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

DEMO

" Slackspace Evidence"

Searching for evidence in slackspaces

We cant 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.

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
```


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

"File Carving Techniques"

Analysis of unallocated areas that may contain relevant artifacts.

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.

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

Recovery

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

The expert needs to use specific tools for each type of filesystem

Useful tools for recovery

- **Magicrescue** – together with DLS, this permits the recovery of the files
- **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 image.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

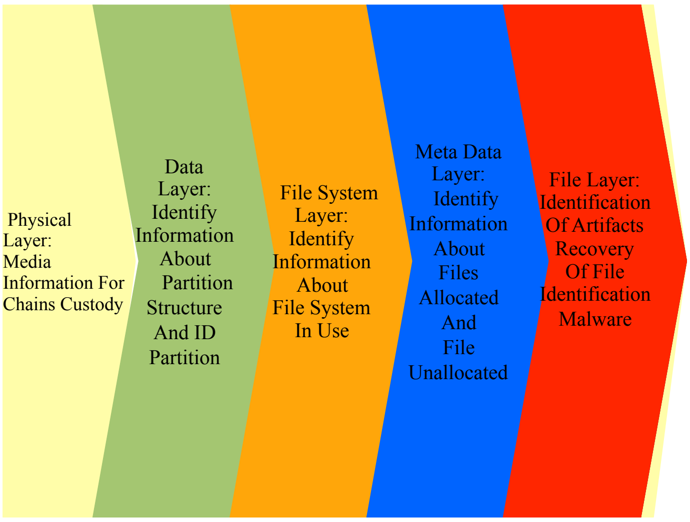
Recovery with Foremost

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

DEMO

This is an example with foremost .

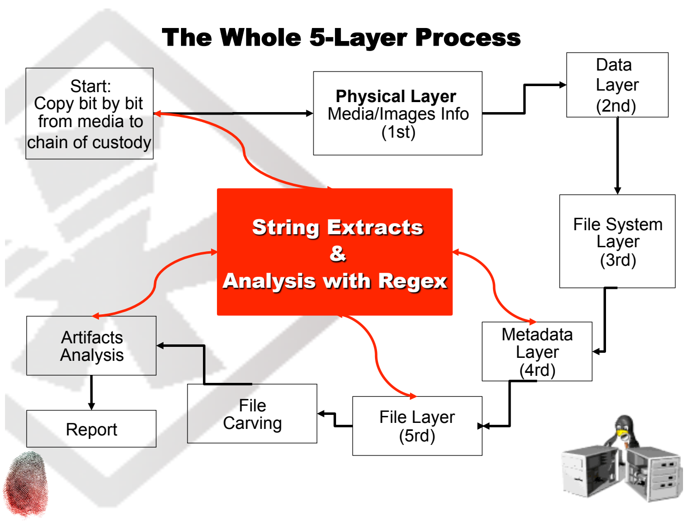


So, there are many tools for the Post Mortem Process.

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



and finally the whole post mortem analysis and its related string analysis

Conclusion

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.

Every Forensic examiner should Compile his own kernel just like
every Jedi builds his own light Saber”

(The Cory Altheide – Google security)

So, there are many FLOSS tools CLIS (Command Line On Steroids) and also GUI Tools (example: Autopsy, Pyflag, PTK) for the Post Mortem Process

Linux OS is the better choice for Computing Forensics, because it supports many file systems 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

ANY QUESTIONS ?

Some beer?

Any Questions

Who’s paying for the beer?