22nd ANNUAL FIRST CONFERENCE MIAMI

JUNE 13-18, 2010

Conference Speaker

The first stage of this tutorial was developed by, Sandro Melo – 4NIX (www.4nix.com.br) and Nelson Uto, with the goal to be a reference in the studies of the Computer Forensic Course, using many tools as FOSS (Free and Open Source Software).

The second stage with Andreas that about Forensic Hand On (really)!

About Sandro Melo

currently working for Locaweb (the biggest hosting company of Latin America) as an Archtecth Linux and Incident Response of Secutiy Member Group, is Proctor of LPI and BSDA certification , has worked for 4NIX as an instructor of Network Security, Pentest and Computer Forensic in courses throughout Brazil, and also is Invited Professor in Lavras University - UFLA (MG), FACID (PI), IBTA College (SP), Portiguar University (RN), Air Force Institute of Technology - ITA (SP), Atual da Amazonia College (RR) and Chair Professor of Operating Systems in Bandtec College (SP).

He holds a master's degree in Network Engineering from Institute Search of Sao Paulo – IPT / USP. He is a writer and technical reviewer, author of four books published in Brazil by Altabooks publisher.

Throughout his career, spanning more than eighteen years, he worked in projects for the biggest companies, as example: IBM, EMC, EDS/HP, banks; many organizations of the Brazilian government and also for militaries organizations.

About Nelson Uto

She has been an Information Technology professional for 13 years and an Information Security specialist for the last 7 years. He currently works at CPqD Telecom & IT Solutions as a Security Consultant and Researcher, in the areas of Cryptography and Application Security, and also as a PCI QSA and a PCI PA-QSA: he worked on cryptographic key management, evaluated free libraries supporting elliptic curve cryptography for the XScale and x86 platforms, performed pentests on several web applications as part of a risk analysis project, prepared hardening guidelines for Oracle and Unix systems, researched the application of K-Means clustering algorithm for semiautomatic generation of security event correlation rules, specified a security event management system, and elaborated security policies.

Introduction

In past, a server 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 FOSS system. Given this degree of risk, at first intangible, the threat of an invasion is something that we can't overlook.

In this context, the forensic techniques are essential during the response to an incident, to identify where the computer has violated its security, what was changed, the identity of the attacker and preparing the environment for expertise of Forensic Computer.

Bearing in mind the care of an expert as a Computer Forensic, invasion is electronic crime. A digital evidence must be preserved so that it can have value.

22nd ANNUAL FIRST CONFERENCE MIAMI

JUNE 13 - 18, 2010

Conference Speaker

BRASIL UM PAÍS DE TODOS

GOVERNO FEDERAL

Sandro Melo

sandro@4nix.com.br

sandro@ginux.ufla.br

Nelson Uto

uto.cseg@gmail.com

First Time :

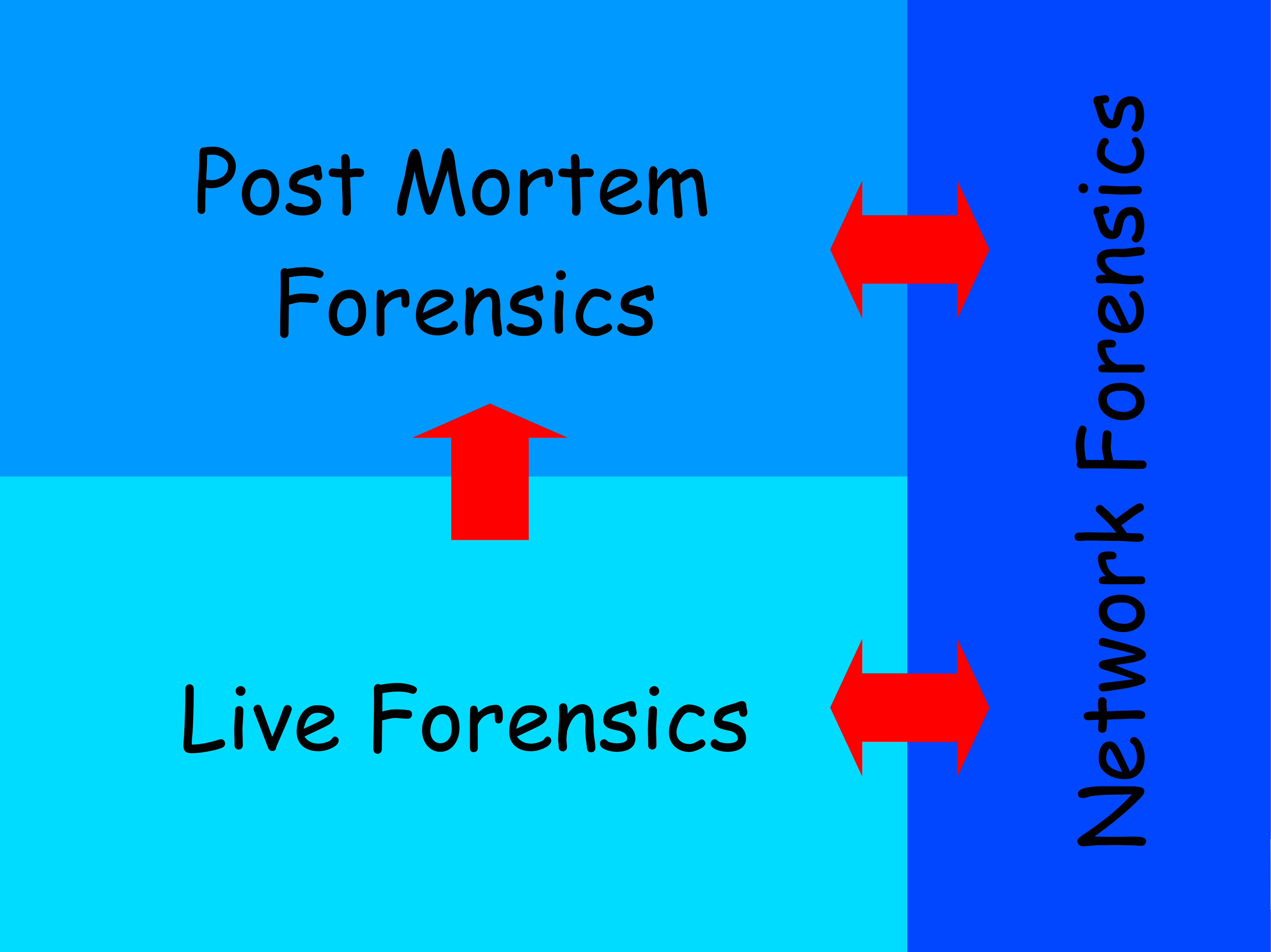
“ HANDS ON

POST MORTEM

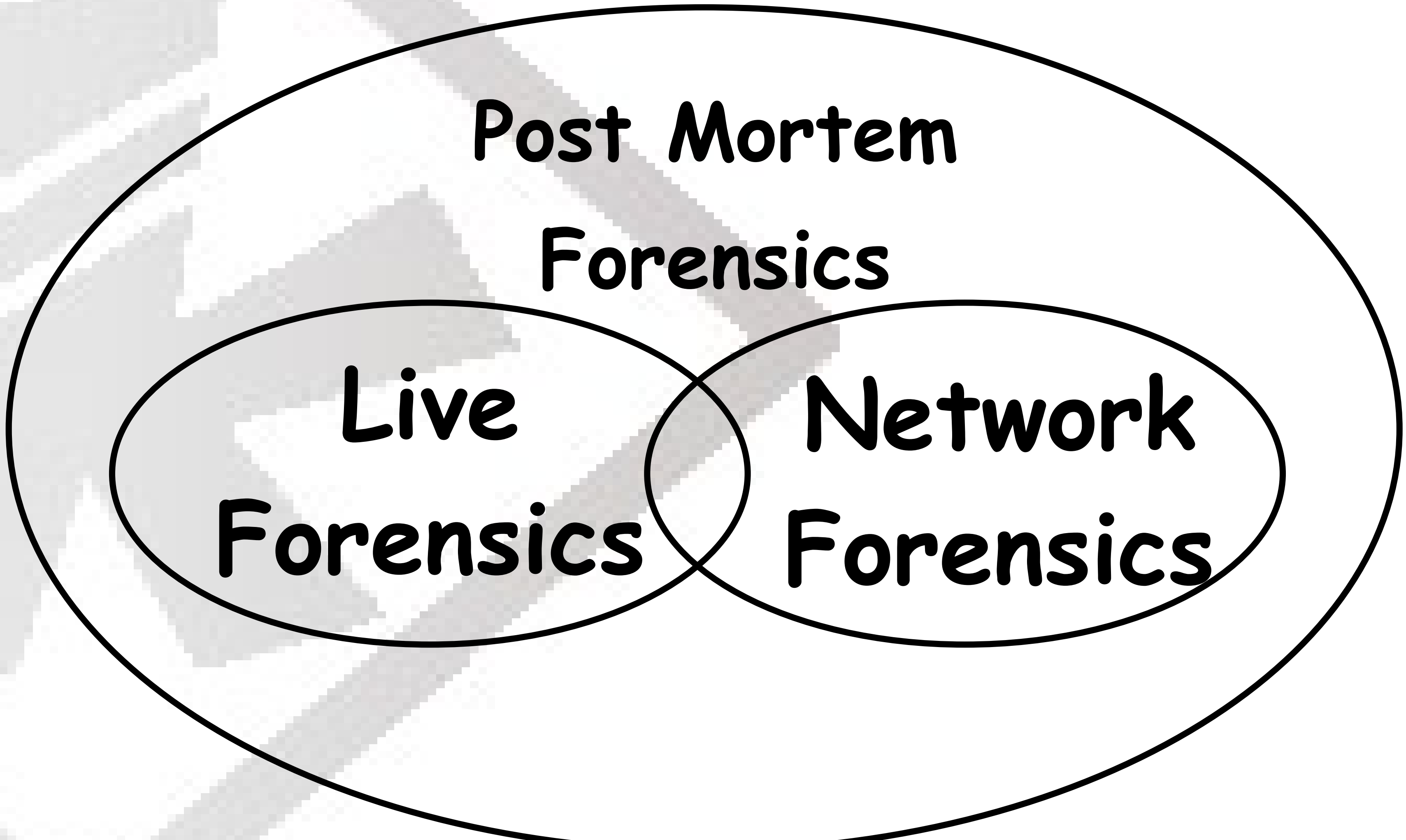
FORENSIC ANALYSIS with specifics Forensic FOSS TOOLS”

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

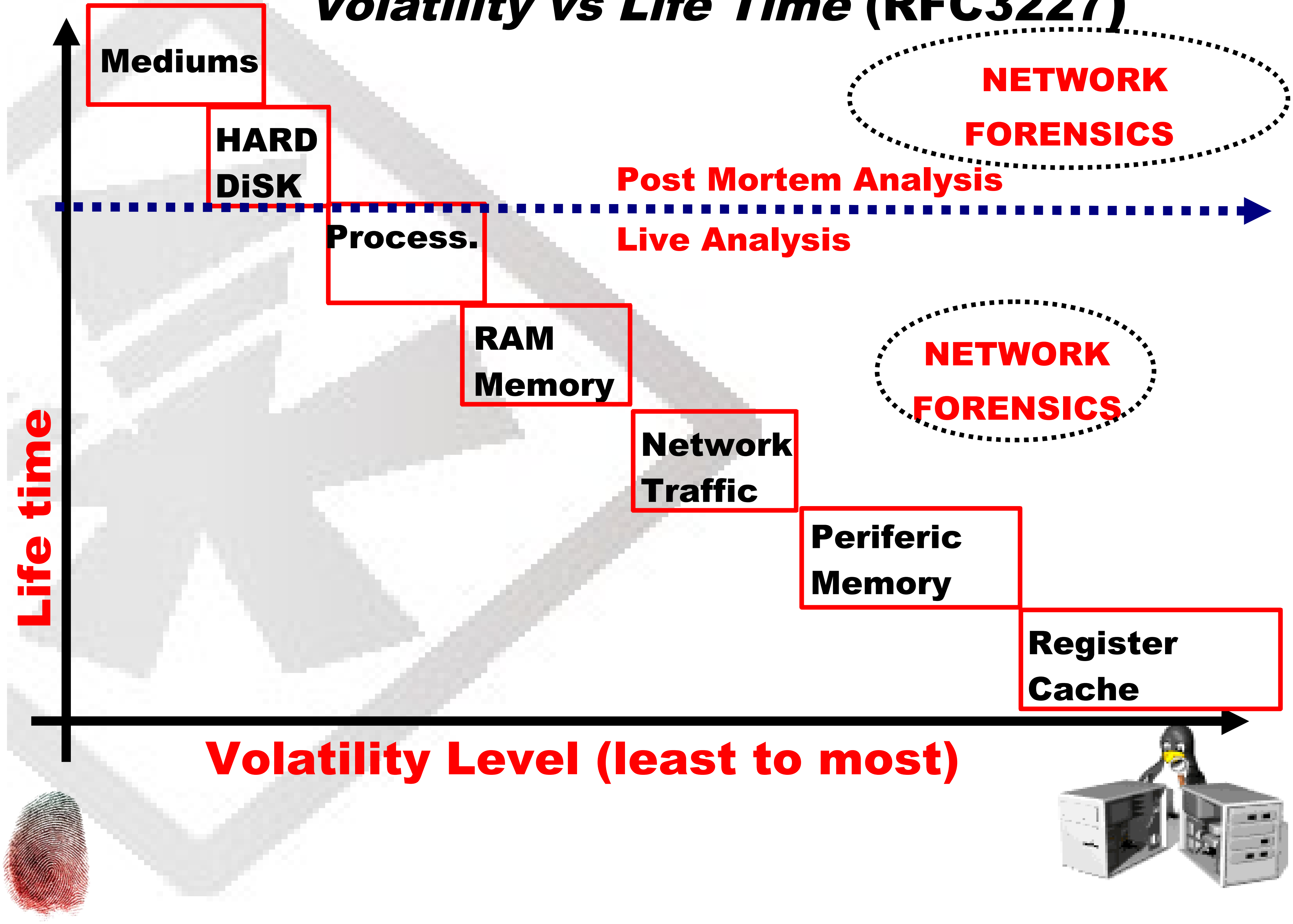
"Initial Concepts"


Post Mortem Forensics
Live Forensics
Network Forensics

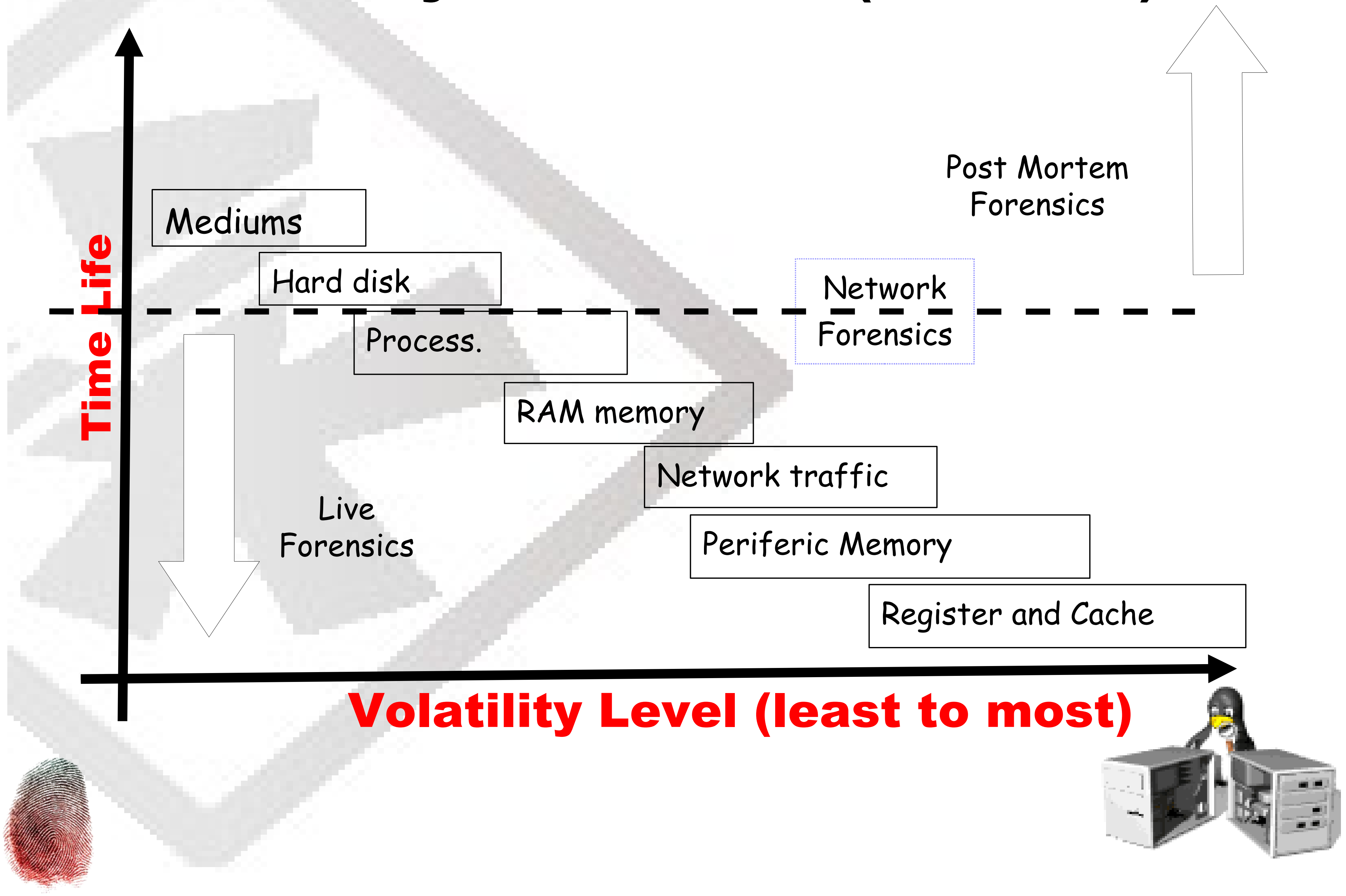
Correlations of Forensic Evidences found.



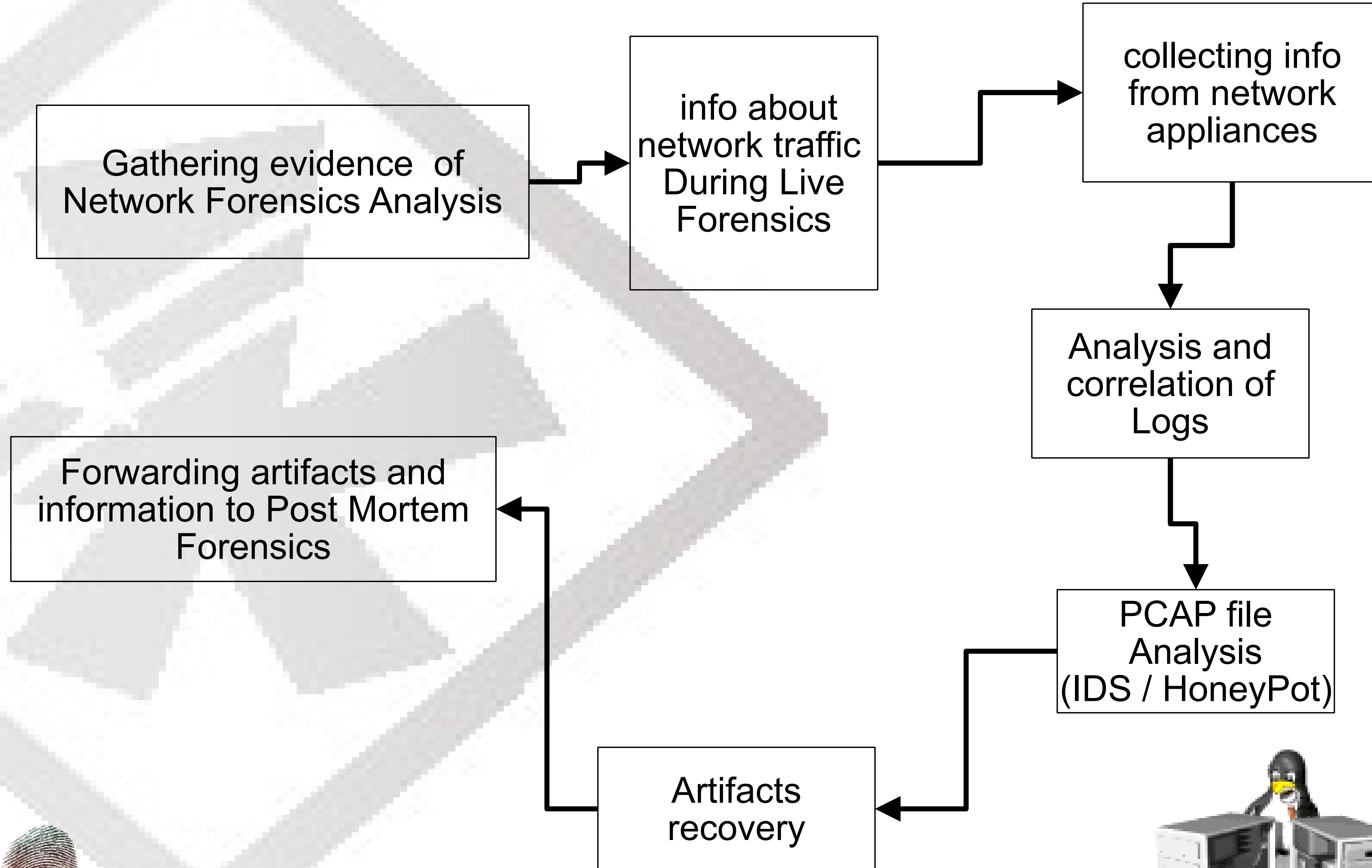
Volatility vs Life Time (RFC3227)



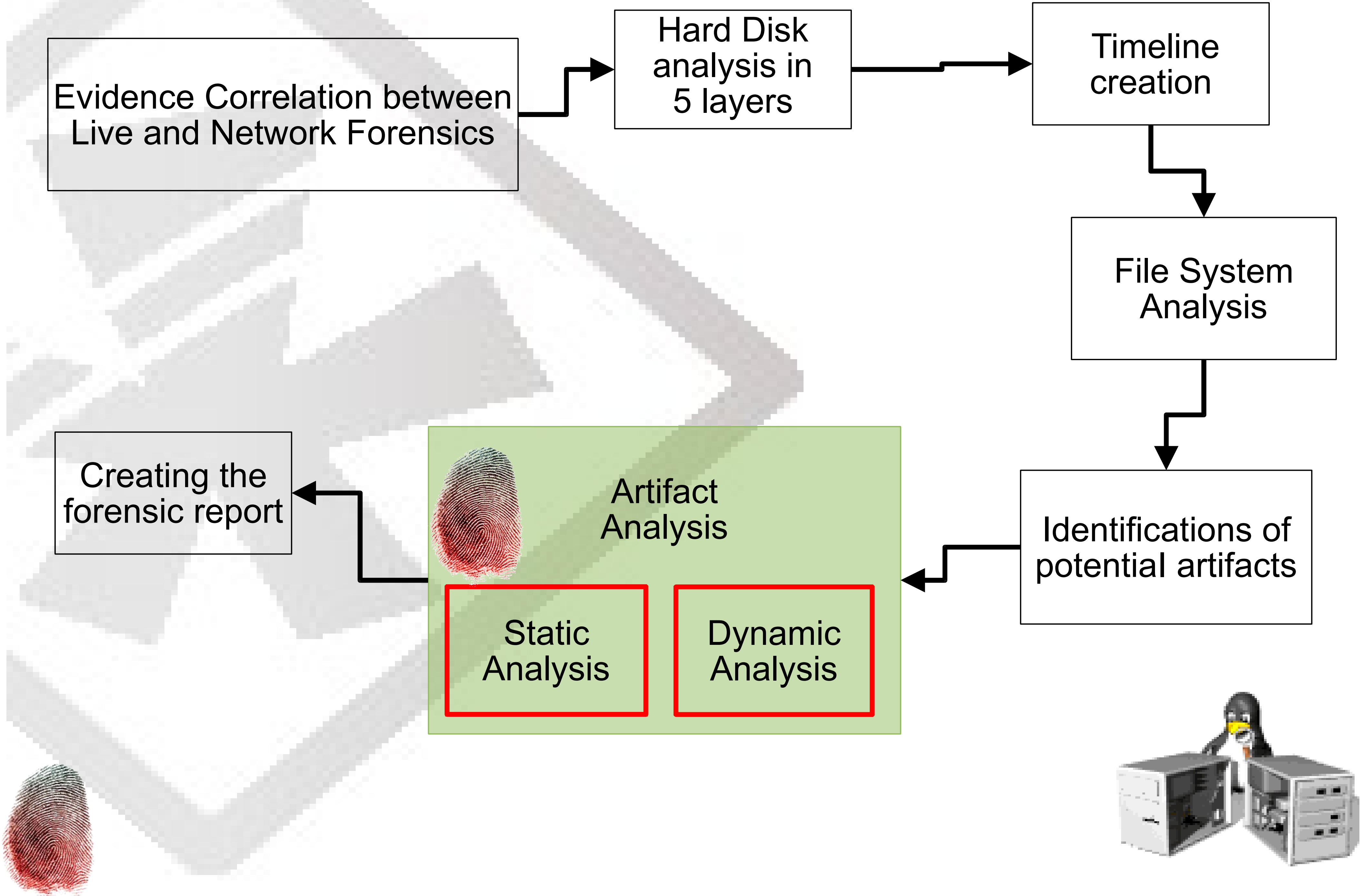
Volatility vs Life Time (RFC3227)



Network Forensics



Post Mortem Analysis



Initial System Analysis

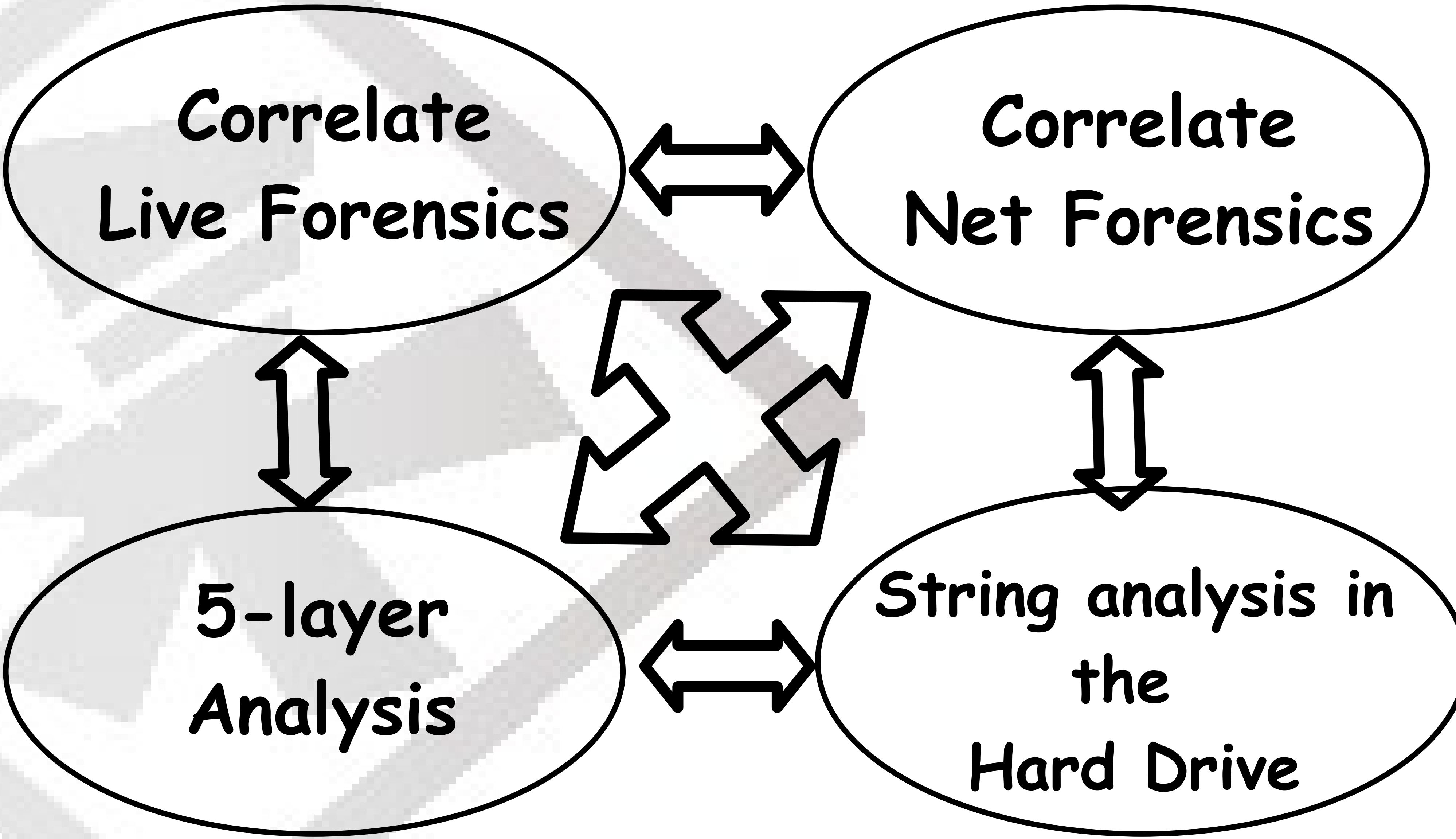
Several actions can be taken in an attempt to find evidence and artifacts related to Security Incidents under investigation.

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

Initial System Analysis

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

Post Mortem – Correlations



Concept

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 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.

Concept

Strings vs Regex

grep -i“tar\.gz$” imagem.string

egrep --regexp=“\.tgz|\.zip|\.bz2|\.rar|\.c” imagem.string

Strings vs Regex

grep -E "[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}" imagem.string

grep -i "\/exploit\/" imagem.string

grep -i "\/exploits\/" imagem.string

grep -i "rootkit\/" imagem.string

grep -i "\/\.\.\ " imagem.string

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
```

Concept

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 automatize 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

Concept

“Media Analysis”

Using the 5-layer concept (Image: Hard drives, USB-drives, flash memory drives ...)

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)

Concept

“Physical Layer”

(Analysis of information from media and/or image)

Physical Layer

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

Hard disk drives

Removable media

Useful information from Media

Useful information that, in most cases, has already been collected during Live Forensics.

```
# cat /proc/partitions
# hdparm -i /dev/hda
# hdparm -I /dev/hda
```

“Data Layer”

(Analysis of information from boot sector and partitioning)

Data Layer

A preliminary step for this phase of the analysis happens when information is gathered from 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.

Useful Tools to Data Layer

It collects hard disk basic info:

- disk_stat
- disktype
- file
- scsiinfo

It shows partition info from HD or image:

- fdisk
- sfdisk

It shows partition and slackspace info from HD or image:

- mmls

Userful Tools to Data Layer

It allows to see partition info and if necessary to recovery partition structure:

- testdisk

It collects hard disk or image static info:

- img_stat
- mmstat

It allows manipulation of image and HD

- mount
- losetup

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

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
```

Get static info with DISK_STAT from device

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

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
```

Get info with SCSIINFO from device

```
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 with FDISK from image

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

```
# fdisk -lu image.img

# sfdisk -luS image.img
```

Get info with FDISK from device

```
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

Get info with FDISK from image

```
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

Get info with SFDISK from device

```
# 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

Get info with MMLS from device

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

Get info with MMLS from image

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

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
```

"Filesystem Layer"

(To use in file system structure analysis)

Useful Tools for File System Layer

Common tools to collect info from the File system

- Fsstat

It gets journaling info from image, (e.g. statistics info about partition)

- jcat

It shows general info from journaling file system

- jls

It shows journaling info from strucutre of file system

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,
```

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

```
# 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
```

"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 to search for evidence related to the incident.

The metadata analysis information is an extremely important step in the search for evidences and other actions in the fifth layer (File Layer).

Useful Metadata Tools

- istat (static info)
- ils
- ifind

It shows Inode structure info

- icat

It collects content of a specific Inode

- mactime

It collects mactime info of all files in the Inode table and allows to create the timeline.

The all important timeline

It's a big report with all files info and its mactime:

The timeline is created based on MACtime (Modified, Accessed, Created/Changed)

Info of when:

- the Operation system (O.S.) was installed.
- Changes and updates were made
- the O.S. was used for the last time
- and many other details related to the manipulated filesystem's files.

Sleuthkit Timeline creation

Example of how to create 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
```

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

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

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

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
```

“File Layer”

(Analysis of file information and identification of possible artifacts)

Data Blocks useful tools

- dstat

shows statistic info from data blocks

- dls

enables to list info from allocated, unallocated and slackspace areas

- dcat
- dcalc

manipulate info from a specific data block

File Layer

Tools for File Layer analysis

fls

Enables one to consult file and directory info from an image.

Ffind

Similar to fls but using the specific Inode address.

Sorter

Enables to sort the files according to its type.

“Image Mounting”

Image Mounting

It's recommended that disk forensic image analysis be a process executed with caution, begining 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.

Example on 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)
```

Example on image mounting of multiple partitions

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

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

Example on 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 initial sector of the partition is 73. Considering this case, this value must be multiplied by 512 to calculate of offset value.

Expr 73 * 512

The result determining the offset value is **37376**

Mouting a partition from the full disk image

Previous to 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
```

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

Preparation for mounting of partition with losetup

```
# losetup -a

# expr 410319 \* 512

210083328

# losetup -o 210083328 /dev/loop2 HD_coleta.img
```

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
# 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

It shows info partition mounted

```
# 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
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 kind

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

For this task, SORTER command is the recommended tool.

Using sorter and losetup commands together

Here, an example of the use of 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:

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

Search for artifacts with FIND

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

 # find /img/ -name "*[ ]*" ;

Search for artifacts with FIND

Search for files with no owner or specified group, that can be installed in the system unconventionaly:

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

File Layer

Search for artifacts with FIND

Search for hidden files and directories, 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 invasor:

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

Search for artifacts with FIND

Many invasors 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

Search for artifacts with FIND

Searching for files that are with 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

Searching for artifacts with FIND

Searching for files whose access time within determined time frame. This kind of search is also useful for artifacts 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

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/

Searching Malware

To search Malware info with the command rkhunter:

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

File Layer

Searching Malware

searching for Malware info with "clamav"command:

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

“ Slackspace Evidences”

Searching evidences in slackspaces

File Layer

Searching Slackspace

It is recommended 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).

Periciando Slackspace

It allows to get information about slackspace from image

dls -s image.img | slackspace.dls

strings -a slackspace.dls > slackspace.dls.strings

File Layer

"File Carving Techniques"

Analysis in unallocated areas that may contain relevant artifacts.

File Layer

Recovery

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

Luckly, an Expert has several options when it comes to FOSS tools.

Recovery

Another relevant point is the fact that some file systems not only perform the unlink with the metadata and the data, but also overwrite the metadata with zeroes.

Example: EXT3

Useful tools for recovery

Magicrescue - together with DLS, it permits the recovery of the files

foremost - it recovers files from their headers and footers.

ddrescue - it recovers files from the image of any media.

Recovery using classic procedure

Attempting to recover a file from an image:

a) Identify the addresses (inodes)

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

b) Retrieve the content from list (data)

 # cat list.image.txt

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

 # icat image.img 4157 > file.ppt

Recovery with Foremost

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

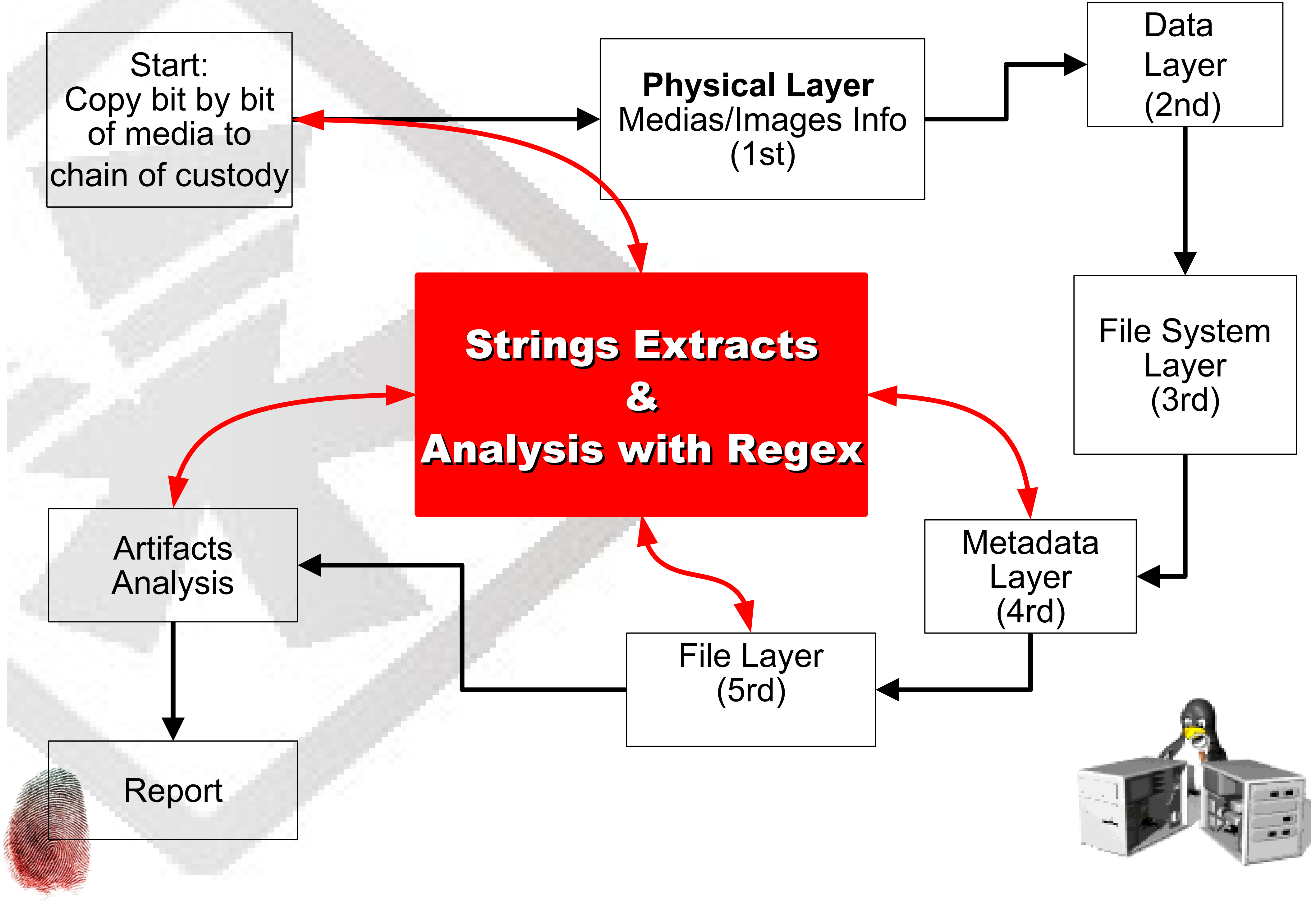
 # foremost -c foremost.conf -i image.img -o /recovery -T

Recovery with Foremost

Another way to use FOREMOST is to perform a search for kind of files. 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

All 5-Layers Process



Conclusion

So, there are many tools to do Post Mortem Process and same we use automated tools, have the vision in "5 Layers " to permit to do an analysis with more details, and also when the tools avaliable are not able to help, and we need to do the analysis of way "hands on".

ANY QUESTION ?