Digital Forensics

An introduction into Post-mortem Digital Forensics



CIRCL
Computer Incident
Response Center
Luxembourg

CIRCL *TLP:WHITE*

info@circl.lu

Version 1.0.2 2018 edition

Overview

1. Introduction
2. From data to knowledge
3. Disk Acquisition
4. Disk Analysis
5. Carving
6. File System Analysis
7. Analysing files
8. String Search
9. Windows Registry Analysis
10. Memory Forensics
11. Outlook



CIRCL
Computer Incident
Response Center
Luxembourg

1. Introduction

1.1 Common default behaviour

- Re-install,

  Re-image,

  Restore from backup

  $\rightarrow$ Destruction of evidences

- Run AV,

  Apply updates,

  Do some investigations

  $\rightarrow$ Create big noise
  $\rightarrow$ Overwrite evidences
  $\rightarrow$ Destruction of evidences

1.2 Answers

→ Was the system compromised

→ How, when, why

→ Malware involved

→ Persistence mechanisms

→ RAT involved

→ Social engineering attack

→ System used for lateral movement

→ System attacked from the LAN

→ System involved at all

→ Access to sensitive data

1.3 Forensics Science

- Forensic classical

  Locard's exchange principle

  `https://en.wikipedia.org/wiki/Locard%27s_exchange_principle`
- Write down everything you see, hear, smell and do
- Chain of custody

  → `https://www.nist.gov/sites/default/files/documents/2017/04/28/Sample-Chain-of-Custody-Form.docx`
- Scope of the analysis

1.4 Forensic disciplines

- Reverse Engineering
- Memory Forensics
  → `https://www.circl.lu/pub/tr-30/`
- Network Forensics
- Mobile Forensics
- Cloud Forensics
- Post-mortem Analysis

1.5 First Responder: physical evidence

- Crime scene:
  - Tower, desktop, laptop, tablets
  - Screen, printer, storage media
  - Router, switches, access point
  - Paper, notes, ...
- Be ready:
  - Photo camera
  - Flash light, magnifying glasses
  - Labelling device, labels, tags, stickers
  - Toolkit, screwdriver kits
  - Packing boxes, bags, faraday bag
  - Cable kits, write blocker, storage devices
  - Anti-static band, network cables
  - Pens, markers, notepads

    $\rightarrow$ Chain of custody

1.5 First Responder: order of volatility

Registers, cache → nanoseconds

RAM memory → tens of nanoseconds

Network state → milliseconds

Processes running → seconds

Disk, system settings, data → minutes

External disks, backup → years

Optical storage, printouts → tens of ears

→ `https://www.circl.lu/pub/tr-22/`

1.5 First Responder: live response

- Powered-on versus powered-off
- Essential system information
  - → System time
  - → Logged-on users
  - → Open files
  - → Network -connections -status
  - → Process information -memory
  - → Process / port mapping
  - → Clipboard content
  - → Services
  - → Command history
  - → Mapped drives / shares
- Memory dump

1.6 Post-mortem Analysis

- Hardware layer & acquisition
  - Best copy (in the safe)
  - Working copy (on a NAS)
  - Disk volumes and partitions
  - Simple tools: dd, dmesg, mount
- Data layer
  - Carving: foremost, scalpel, testdisk/photorec
  - String search
- File system layer
  - FAT, NTFS
  - File system timeline
  - Restore deleted files

1.7 Post-mortem Analysis

- OS layer
  - Registry
  - Event logs
  - Volume shadow copies
  - Prefetch files
- Identify malware
  - TEMP folders
  - Startup folders
  - Windows tasks
- Application layer
  - AV logs
  - Browser history: IE, firefox, chrome
  - Email
  - Office files & PDFs

1.8 Forensic Distributions

- Commercial
  - EnCase Forensic
  - F-Response
  - Forensic Toolkit
  - Helix Enterprise
  - X-Ways Forensics
  - Magnet Axiom

- Open source tools
  - Kali Linux
  - SANS SIFT
  - Digital Evidence and Forensics Toolkit - DEFT
  - PlainSight
  - Computer Aided INvestigative Environment - CAINE



CIRCL
Computer Incident
Response Center
Luxembourg

2. From data to knowledge

2.1 Data in a binary system

- Binary digit $\rightarrow$ BIT
- Data represented as binary patterns

  Ordered sequence

  `x Bits --> 01010000011010010110111001100111 --> y Bits`

  Bit x + 2 = 1

  Bit x + 3 = 0

- Addressing added

```
01010000 01101001 01101110 01100111 -->
-------- -------- -------- --------
Byte 0   Byte 1   Byte 2   Byte 3   -->
```

- Apply interpretative rules on addresses

2.1 Data in a binary system

- Nibble

  `0101 0000 0110 1001 0110 1110 0110 0111`
- Byte

  `01010000  01101001  01101110  01100111`
- Word

  `0101000001101001  0110111001100111`
- Double Word
- Big / Little Endian
- Integer / Signed Integer
- Floating Point
- Binary Coded Decimal
- ASCII, Unicode
- GIF / JPEG / PNG / EXE / ...
- ...

2.2 Example: Integer Bytes

```
0101 0000 0110 1001 0110 1110 0110 0111
---------


0101 0000
|||| ||||__ 0 * 2^0 =   0
|||| |||___ 0 * 2^1 =   0
|||| ||____ 0 * 2^2 =   0
|||| |_____ 0 * 2^3 =   0
||||_______ 1 * 2^4 =  16
|||________ 0 * 2^5 =   0
||_________ 1 * 2^6 =  64
|__________ 0 * 2^7 =   0
                      ---
                       80
```

2.3 Example: Signed Integer Bytes

```
1011 1111                  Two's complement:
 011 1111                      1. Remove the sign
 100 0000                      2. Invert
 100 0001                      3. Add 1
 ||| ||||__ 1 * 2^0 =   1
 ||| |||___ 0 * 2^1 =   0
 ||| ||____ 0 * 2^2 =   0
 ||| |_____ 0 * 2^3 =   0
 |||_______ 0 * 2^4 =   0
 ||________ 0 * 2^5 =   0
 |_________ 1 * 2^6 =  64
                       ---
                       -65
```

2.3 Example: Signed Integer Bytes

Exercise:

```
1101 1100                Two's complement:
                             1. Remove the sign
                             2. Invert
                             3. Add 1
 |||  ||||__    * 2^0 =
 |||  |||___    * 2^1 =
 |||  ||____    * 2^2 =
 |||  |_____    * 2^3 =
 |||_______     * 2^4 =
 ||________     * 2^5 =
 |_________     * 2^6 =
                        ---
```

2.3 Example: Signed Integer Bytes

Exercise:

```
1101 1100                Two's complement:
 101 1100                    1. Remove the sign
 010 0011                    2. Invert
 010 0100                    3. Add 1
 ||| ||||__ 0 * 2^0 =   0
 ||| |||___ 0 * 2^1 =   0
 ||| ||____ 1 * 2^2 =   4
 ||| |_____ 0 * 2^3 =   0
 |||_______ 0 * 2^4 =   0
 ||________ 1 * 2^5 =  32
 |_________ 0 * 2^6 =   0
                       ---
                       -36
```

2.4 Big Endian and Little Endian

Big Endian

```
2^15  2^13  2^11  2^9     2^7   2^5   2^3   2^1
   2^14  2^12  2^10  2^8     2^6   2^4   2^2   2^0
-  -  -  -  -  -  -  -    -  -  -  -  -  -  -  -
0  0  0  1  1  0  0  0    0  1  0  1  0  1  0  1
-  -  -  -  -  -  -  -    -  -  -  -  -  -  -  -
      Byte 10.000              Byte 10.001
```

Little Endian

```
2^7   2^5   2^3   2^1     2^15  2^13  2^11  2^9
   2^6   2^4   2^2   26 0    2^14  2^12  2^10  2^8
-  -  -  -  -  -  -  -    -  -  -  -  -  -  -  -
0  1  0  1  0  1  0  1    0  0  0  1  1  0  0  0
-  -  -  -  -  -  -  -    -  -  -  -  -  -  -  -
      Byte 10.000              Byte 10.001
```

2.4 Big Endian and Little Endian

Exercise: Interpret the bit sequence as words

```
10010110 10100101 00001111 11000011
-------- -------- -------- --------
```

Big Endian:

```
-------- --------    -------- --------
```

Little Endian:

```
-------- --------    -------- --------
```

2.4 Big Endian and Little Endian

Results: Interpret the bit sequence as words

```
10010110 10100101 00001111 11000011
-------- -------- -------- --------
```

Big Endian:

```
10010110 10100101    00001111 11000011
-------- --------    -------- --------
```

Little Endian:

```
10100101 10010110    11000011 00001111
-------- --------    -------- --------
```

2.5 From Bin to Hex

Example:

```
0001 1000     0101 0101     0000 1111     1010 0110
---------     ---------     ---------     ---------
   0x18          0x55          0x0F          0xA6
```

Exercise:

```
1001 0110     1010 0101     0000 1111     1100 0011
---------     ---------     ---------     ---------
   0x            0x            0x            0x
```

2.5 From Bin to Hex

Exercise:

```
1001 0110     1010 0101     0000 1111     1100 0011
---------     ---------     ---------     ---------
   0x            0x            0x            0x
```

Results:

```
1001 0110     1010 0101     0000 1111     1100 0011
---------     ---------     ---------     ---------
   0x96          0xA5          0x0F          0xC3
```

2.6 Example: Others

Packed BCD

```
0110 1110 0110 0111
---- ---- ---- ----
  6   na    6    7
```

ASCII

```
0101 000 0110 1001 0110 1110 0110 0111

01010000  01101001  01101110  01100111
--------  --------  --------  --------
     80        105       110       103
      P          i         n         g
```

2.7 Data, files, context

- Sequence of data & interpretative rules $\rightarrow$ file
- File $\rightarrow$ name & description
- Files $\rightarrow$ File system

- Where did you find the string "ping"?
  - Binary inside TEMP folder
  - Autorun folder
  - Registry
  - Browser history
  - Command line history

  $\rightarrow$ Data $\rightarrow$ Information $\rightarrow$ Knowledge



CIRCL
Computer Incident
Response Center
Luxembourg

3. Disk Acquisition

3.1 Storage devices / media

- IBM 305 RAMAC
  - Random Access Method of Accounting and Control
  - 1956
- IBM 350 Disk Storage
  - 152 x 172 x 63 cm
  - 50.000 blocks of 100 Characters $\rightarrow$ 5MB

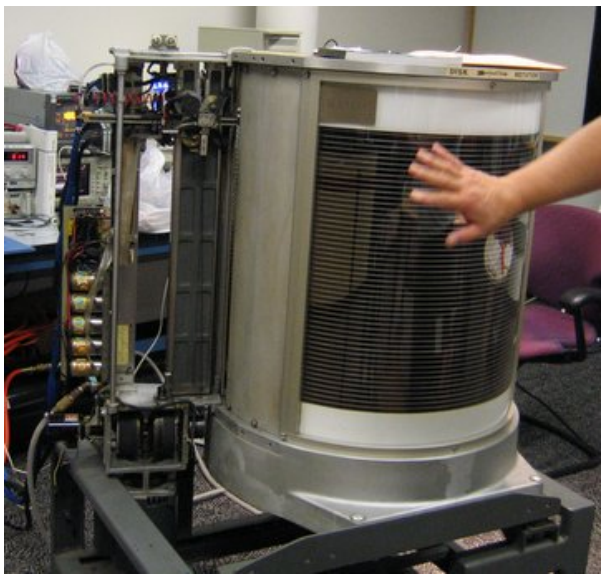


Image (c) wikipedia.org - Image used solely for illustration purposes

3.1 Storage devices / media

- Magnetic tapes
- Floppy disks
  - 8" - 1971 - 80KB
  - 5.25" - 1976 - 360 KB
  - 3.5" - 1984 - 1.2 MB / - 1986 - 1.44 MB
- Optical storage media
  - Compact disks - CD
  - Digital versatile disk - DVD
  - Blu-ray disk
- Flash storage media
  - USB flash drive
  - Flash memory cards
- Hard disks
  - IDE / EIDE HDDs
  - SATA HDDs
  - SSDs

3.2 ATA Disks

- ATA-3
  - Hard disk password
- ATA-4
  - Host Protected Area - HPA
  - Vendor area
  - `READ_NATIVE_MAX_ADDRESS`
- ATA-6
  - Device Configuration Overlay - DCO
  - Optional features of a HD
  - `DEVICE_CONFIGURATION_IDENTIFY`
- ATA-7
  - Serial ATA

3.3 Disk cloning - imaging

- Clone disk-2-disk
  - Different sizes
  - Wipe target disk!
- Clone disk-2-image
  - Clear boundaries
  - One big file
  - Break file into chunks
- Image file format
  - RAW
  - EWF (Additional information)
  - Please no 3rd party formats
- Write-Blockers
  - Hardware

3.4 Linux commands

- `dmesg`

```
[106831.987923] scsi6 : usb-storage 2-1:1.0
[106834.126960] scsi 6:0:0:0: Direct-Access  USB  Flash DISK  1100 PQ: 0 ANSI: 4
[106834.127269] sd 6:0:0:0: Attached scsi generic sg1 type 0
[106834.127503] sd 6:0:0:0: [sdb] 15826944 512-byte logical blocks: (8.10 GB/7.54 GiB)
[106834.130380] sd 6:0:0:0: [sdb] Write Protect is off
```

- `fdisk -l /dev/sdb`

```
Disk /dev/sdb: 8103 MB, 8103395328 bytes
250 heads, 62 sectors/track, 1021 cylinders, total 15826944 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x84a18fad

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     4196351     2097152    7  HPFS/NTFS/exFAT
/dev/sdb2         4196352     8390655     2097152    7  HPFS/NTFS/exFAT
/dev/sdb3         8390656    14682111     3145728    7  HPFS/NTFS/exFAT
/dev/sdb4        14682112    15826943      572416   83  Linux
```

3.4 Linux commands

- `mount`

```
mount
umount /media/root/part2_NTFS


mkdir /media/root/NTFS
mkdir /media/root/FAT


mount -o ro,noload /dev/sda2 /media/root/NTFS


losetup /dev/loop6 /dev/sda
mount -o ro,noload,offset=$((512*4196352)) /dev/loop6 /media/root/NTFS


mount -o remount,ro /dev/sda2 /media/root/NTFS
```

3.5 dd

```
$ dd if=example-01.txt of=out-01.txt bs=512

     <input file>      <output file><block size>
                                      (default)
3+0 records in
3+0 records out
1536 bytes (1.5 kB) copied, 0.000126 s, 12.2 MB/s


$ ll
-rw-rw-r-- 1 hamm hamm 1536 May 16 16:51 example-01.txt
-rw-rw-r-- 1 hamm hamm 1536 May 16 17:16 out-01.txt


$ dd if=example-02.txt of=out-02.txt bs=512
3+1 records in
3+1 records out
1591 bytes (1.6 kB) copied, 0.00016048 s, 9.9 MB/s


$ ll
-rw-rw-r-- 1 hamm hamm 1591 May 16 17:10 example-02.txt
-rw-rw-r-- 1 hamm hamm 1591 May 16 17:23 out-02.txt
```

3.5 dd

Exercise: dd

```
dd if=example-02.txt bs=512 skip=0 count=1 | less
dd if=example-02.txt bs=512 skip=1 count=1 | less
dd if=example-02.txt bs=512 skip=2 count=1 | less
dd if=example-02.txt bs=512 skip=3 count=1 | less

dd if=example-02.txt bs=1 skip=0 count=14 | less
dd if=example-02.txt bs=14 skip=0 count=1 | less
```

Exercise: dd | xxd

```
$ dd if=example-02.txt bs=512 skip=3 count=1 |xxd
        0+1 records in
        0+1 records out
        55 bytes (55 B) copied, 5.04e-05 s, 1.1 MB/s

0000000: 4f76 6572 6865 6164 2031 3233 3435 3637  Overhead 1234567
0000010: 3839 3020 204d 6573 7361 6765 2d31 2020  890  Message-1
0000020: 3039 3837 3635 3433 3231 2020 2020 2020  0987654321
0000030: 2020 2020 2020 20
```

3.5 dd

Exercise: Error handling

```
$ dd if=example-02.txt of=out-03.txt bs=512 conv=noerror,sync
        3+1 records in
        4+0 records out
        2048 bytes (2.0 kB) copied, 0.000506747 s, 4.0 MB/s


$ xxd out-03.txt |less
```

Exercise: '&' and 'kill -10'

Exercise: Use 'dd' to restore one file out file out of chunked files.

3.6 Disk acquisition

- Tools
  - dd
  - ddrescue
  - DC3DD - Department of Defense Cyber Crime Center
  - DCFLDD - Defense Computer Forensic Labs
  - Guymager
- `md5sum /dev/sdb`

  → `1822eb36c723c5e174621fefcfe9b1ef`
- `sha1sum /dev/sdb`

  → `9db6f71d1fb79174d0aa1ab9bd0e83d990a4f79b`
- dc3dd features
  - On the fly hashing & meta data
  - Error handling & support forensic formats
  - Splitting output files & verification of files

3.7 Exercise: dc3dd

```
dc3dd if=/media/root/part2_NTFS/dd/img_1.raw  -/          # Input file
      log=img_1.log  -/                                   # Lgging
      hash=md5 hash=sha1  -/                              # Hashing
      ofsz=512 ofs=out_1.dd.000                           # Chunk files 512 Byte

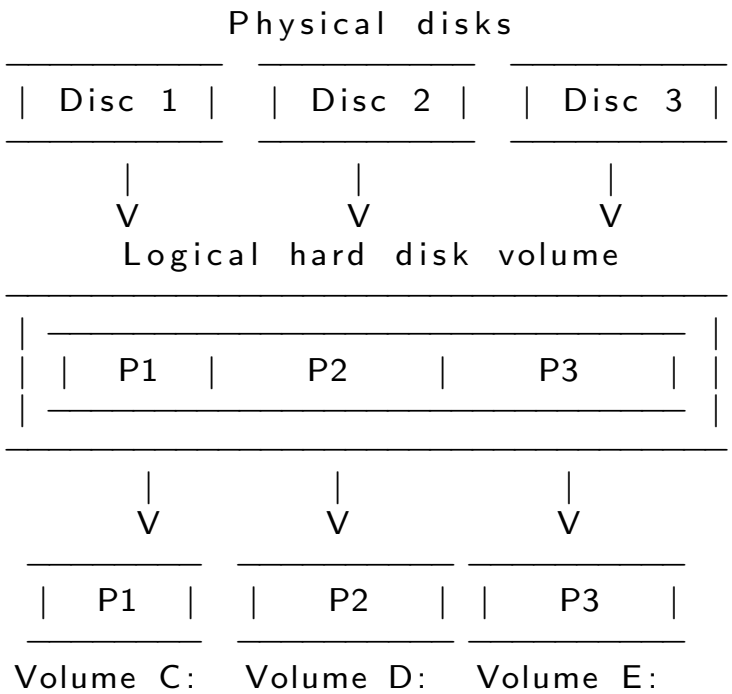
ls -l

less img_1.log

cat out_1.dd.00* | md5sum                                 # Verify hashes
cat out_1.dd.00* | sha1sum

dc3dd wipe=/dev/sdx                                       # Wipe a drive
```

3.8 Physical- / Logical layers





CIRCL
Computer Incident
Response Center
Luxembourg

4. Disk Analysis

4.1 MBR - Master Boot Record

```
# dd if=/dev/sdc bs=512 count=1 skip=0 |xxd

0000000: fab8 0010 8ed0 bc00 b0b8 0000 8ed8 8ec0   ................
0000016: fbbe 007c bf00 06b9 0002 f3a4 ea21 0600   ...|.........!..
0000032: 00be be07 3804 750b 83c6 1081 fefe 0775   ....8.u........u
0000048: f3eb 16b4 02b0 01bb 007c b280 8a74 018b   .........|...t..
0000064: 4c02 cd13 ea00 7c00 00eb fe00 0000 0000   L.....|.........
0000080: 0000 0000 0000 0000 0000 0000 0000 0000   ................
0000096: 0000 0000 0000 0000 0000 0000 0000 0000   ................
...
...
0000432: 0000 0000 0000 0000 9af0 0200 0000 0020   ...............
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8   !...........%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 001a   .....'.X)...]...
0000480: b427 076c dad2 0018 8700 00c0 6800 0000   .'.l........h...
0000496: 0000 0000 0000 0000 0000 0000 0000 55aa   ..............U.
```

000 – 439	0x000 – 0x1B7	Boot code
440 – 443	0x1B8 – 0x1BB	Disc signature
444 – 445	0x1BC – 0x1BD	Reserved
446 – 509	0x1BE – 0x1FD	Partitiontable
510 – 511	0x1FE – 0x1FF	0x55 0xAA

4.1 MBR - Master Boot Record

```
# dd if=/dev/sdc bs=512 count=1 skip=0 |xxd

0000000: fab8 0010 8ed0 bc00 b0b8 0000 8ed8 8ec0   ................
0000016: fbbe 007c bf00 06b9 0002 f3a4 ea21 0600   ...|.........!..
0000032: 00be be07 3804 750b 83c6 1081 fefe 0775   ....8.u........u
0000048: f3eb 16b4 02b0 01bb 007c b280 8a74 018b   .........|...t..
0000064: 4c02 cd13 ea00 7c00 00eb fe00 0000 0000   L.....|.........
0000080: 0000 0000 0000 0000 0000 0000 0000 0000   ................
0000096: 0000 0000 0000 0000 0000 0000 0000 0000   ................
...
...
0000432: 0000 0000 0000 0000 9af0 0200 0000 0020   ...............
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8   !...........%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 001a   .....'.X)...]...
0000480: b427 076c dad2 0018 8700 00c0 6800 0000   .'.l........h...
0000496: 0000 0000 0000 0000 0000 0000 0000 55aa   ..............U.
```

```
Partitiontable:
  Offset: 0      Size: 1 Value: 0x80     --> Bootable
  Offset: 1      Size: 3 Value:          --> Starting CHS address
  Offset: 4      Size: 1 Value: 0x0B     --> FAT32
                                 0x07     --> NTFS
  Offset: 5      Size: 3 Value:          --> Ending CHS address
  Offset: 8      Size: 4 Value:          --> Starting LBA address
  Offset:12      Size: 4 Value:          --> LBA size in sectors
```

4.1 MBR - Master Boot Record

```
...
...
0000432: 0000 0000 0000 0000 9af0 0200 0000 0020   ................
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8   !...........%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 001a   .....'.X)...]...
0000480: b427 076c dad2 0018 8700 00c0 6800 0000   .'.l........h...
0000496: 0000 0000 0000 0000 0000 0000 0000 55aa   ..............U.
```

```
Partitiontable:
  Offset: 0      Size: 1 Value: 0x80     ---> Bootable
  Offset: 1      Size: 3 Value:          ---> Starting CHS address
  Offset: 4      Size: 1 Value: 0x0B     ---> FAT32
                                 0x07     ---> NTFS
  Offset: 5      Size: 3 Value:          ---> Ending CHS address
  Offset: 8      Size: 4 Value:          ---> Starting LBA address
  Offset:12      Size: 4 Value:          ---> LBA size in sectors
```

- Exercise: Calculate the size if the partitions

4.1 MBR - Master Boot Record

```
...
...
0000432: 0000 0000 0000 0000 9af0 0200 0000 0020   ...............
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8   !...........%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 001a   .....'.X)...]...
0000480: b427 076c dad2 0018 8700 00c0 6800 0000   .'.l........h...
0000496: 0000 0000 0000 0000 0000 0000 0000 55aa   ..............U.
```

- Exercise: Calculate the size if the partitions

```
0080 2500     0x 00 25 80 00     2457600     * 512     1258291200   1.2 GB
00c0 5d00     0x 00 5d c0 00     6144000     * 512     3145728000   3.0 GB
00c0 6800     0x 00 68 c0 00     6864896     * 512     3514826752   3.4 GB
```

- Exercise: Change partition type with hexeditor
- Exercise: Find password in unused space before first partition

4.2 VBR - Volume Boot Record

```
# dd if=/dev/sdc1 bs=512 count=1 skip=0 |xxd

0000000: eb58 906d 6b64 6f73 6673 0000 0208 2000  .X.mkdosfs.... .
0000010: 0200 0000 00f8 0000 3e00 f800 0000 0000  ........>.......
0000030: 0100 0600 0000 0000 0000 0000 0000 0000  ................
0000040: 0000 29a2 20e9 9c46 4154 2020 2020 2020  ..). ..FAT
0000050: 2020 4641 5433 3220 2020 0e1f be77 7cac    FAT32   ...w|.
0000060: 22c0 740b 56b4 0ebb 0700 cd10 5eeb f032  ".t.V.......^..2
...
...
00001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

```
0 - 2            Size:  3    Boot code: Where to bot the OS
3 - 10           Size:  8    OEM-ID: mkdosfs
11 - 12          Size:  2    Bytes per sector: 0x0002 -> 0x0200 (little endian)-> 512
13 (0xD)         Size:  1    Sectors per cluster: 0x08 -> 4096 bytes per cluster
50 (0x32) - 51   Size:  2    Boot sector backup: 0x0600 -> 0x0006 -> at sector 6
67 (0x43) - 70   Size:  4    Volume serial number: 0xa220e99c -> 0x9ce920a2
71 (0x47)        Size: 11    Volume label: FAT
82 (0x52)        Size:  8    Partition type: FAT32
90 (0x5A)- 509 (0x1FD)       Not assigned -> boot code & error messages
510 (0x1FE)      Size:  2    Signature: 0x55AA
```

- Exercise: Sleuthkit tools: `nnstat, mmls, fsstat`
- Exercise: Create image of `'carving/formated.dd'`



CIRCL
Computer Incident
Response Center
Luxembourg

5. Carving

5.1 Magic Bytes - File signatures

```
xxd logo_h4k-350x250.jpg | less
0000000: ffd8 ffe0 0010 4a46 4946 0001 0100 0001   ......JFIF......
...
...
0008cc0: 0fa5 0a28 141a 0028 a0d0 3a50 07ff d9     ...(...(..:P...


xxd cases.jpg | less
0000000: ffd8 ffe1 0018 4578 6966 0000 4949 2a00   ......Exif..II*.
...
...
0001730: 4028 0500 a014 0280 501f ffd9             @(......P...


xxd MECO-SMILE.pdf | less
0000000: 2550 4446 2d31 2e34 0a25 c7ec 8fa2 0a35  %PDF-1.4.%.....5
...
...
005c4d0: 3431 390a 2525 454f 460a                  419.%%EOF.


less /etc/scalpel/scalpel.conf
#
#  jpg    y     200000000      \xff\xd8\xff\xe0\x00\x10      \xff\xd9
#
#  pdf    y       5000000      %PDF       %EOF\x0d   REVERSE
#  pdf    y       5000000      %PDF       %EOF\x0a   REVERSE
```

5.2 Carving tools

- Foremost
  - Version 1.5.7
- Scalpel
  - Version 1.60
  - Based on Foremost 0.69
- Bulk Extractor
  - Emails, Email addresses
  - URLs
  - Credit card numbers
  - Social media
  - Telephone numbers
  - ...
- Testdisk - Photorec

5.3 Exercise: Carving deleted data

- Carving deleted data

```
sudo mount -o ro deleted.dd /cdrom/
ls -lh /cdrom/
sudo umount /cdrom

sudo mount -o ro formated.dd /cdrom/
ls -lh /cdrom/
sudo umount /cdrom
```

- Carving deleted data

```
mkdir out1/
foremost -t all -i deleted.dd -o out1/
```

- Carving formatted volume

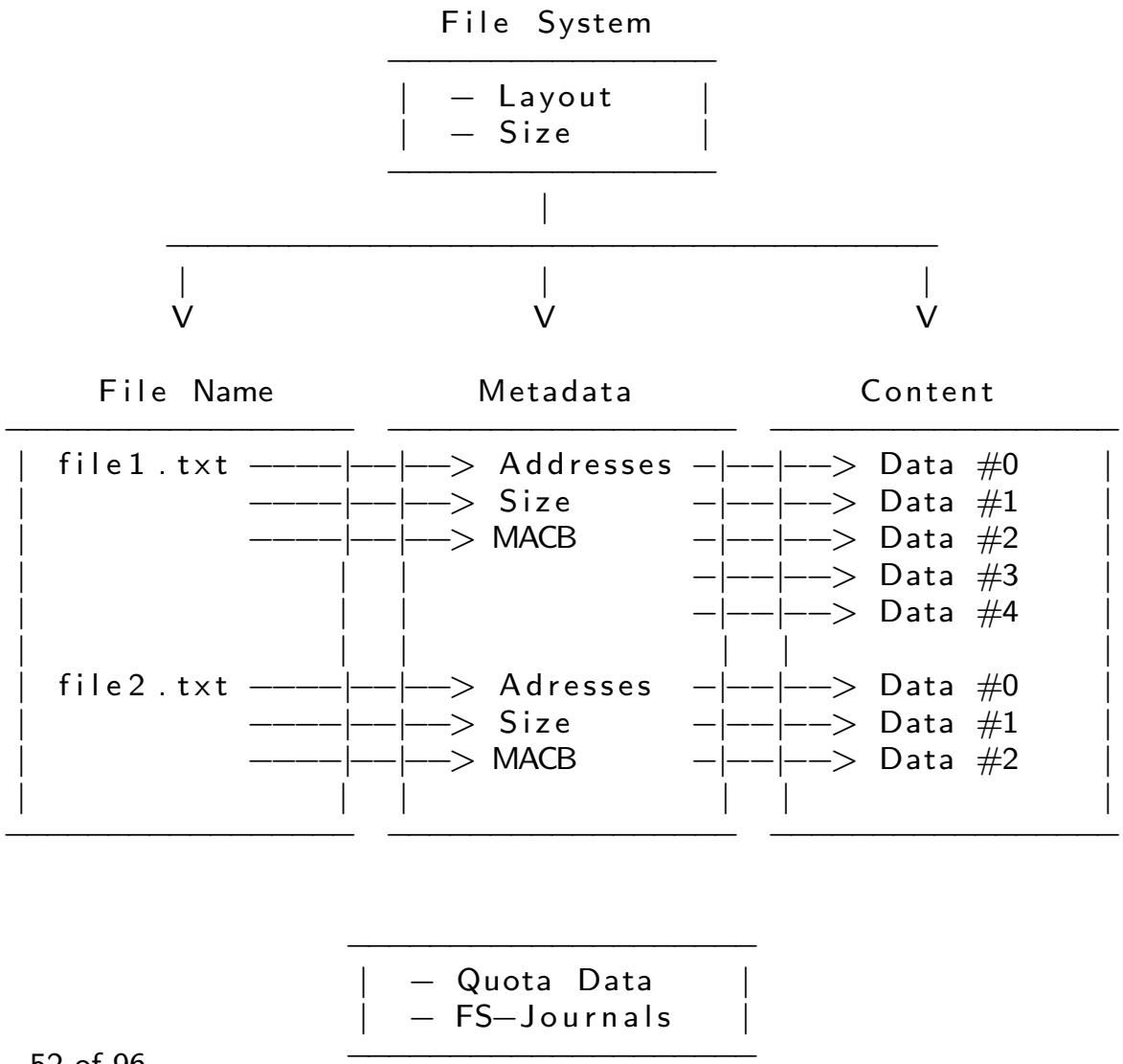
```
mkdir out2/
foremost -t all -i formated.dd -o out2/
```



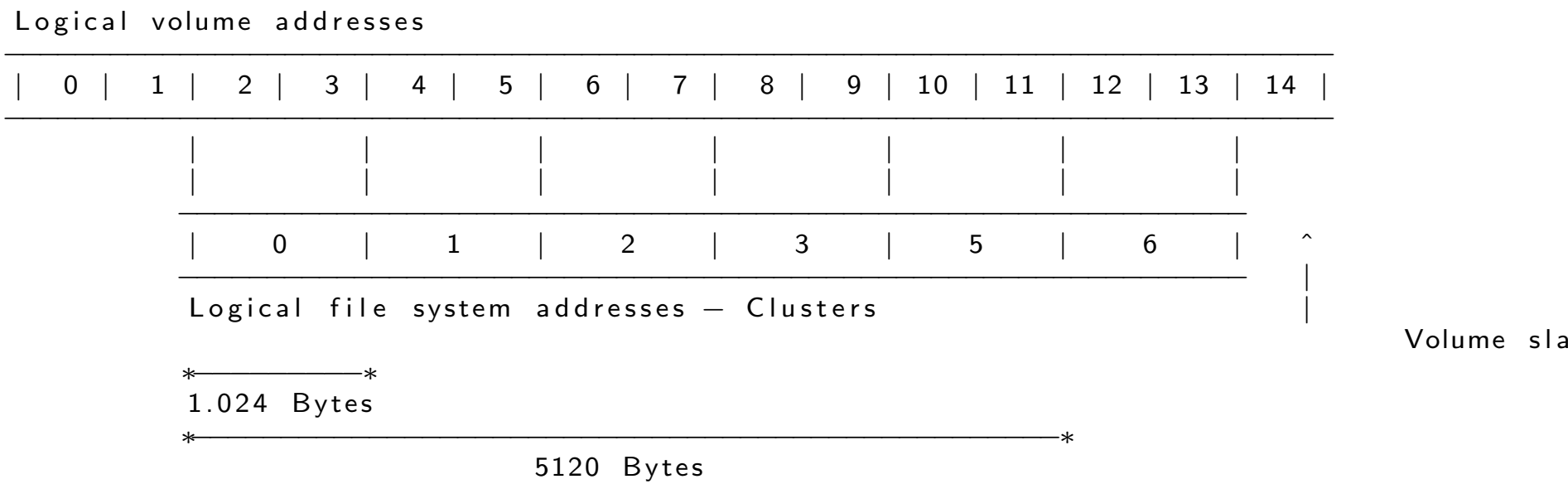
CIRCL
Computer Incident
Response Center
Luxembourg

6. File System Analysis

6.1 Overview: File system components



6.2 Logical file system addresses

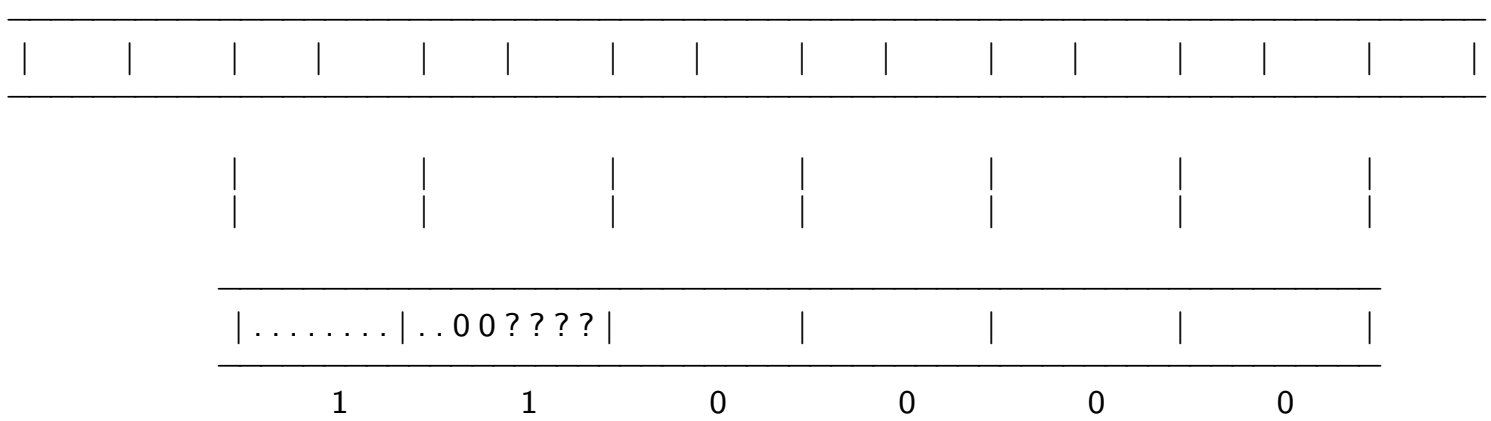


6.3 Allocated / unallocated

....0....	....1....	2	....3....	5	6	7	....8....
1	1	0	1	0	0	0	1

0 = Unallocated
1 = Allocated

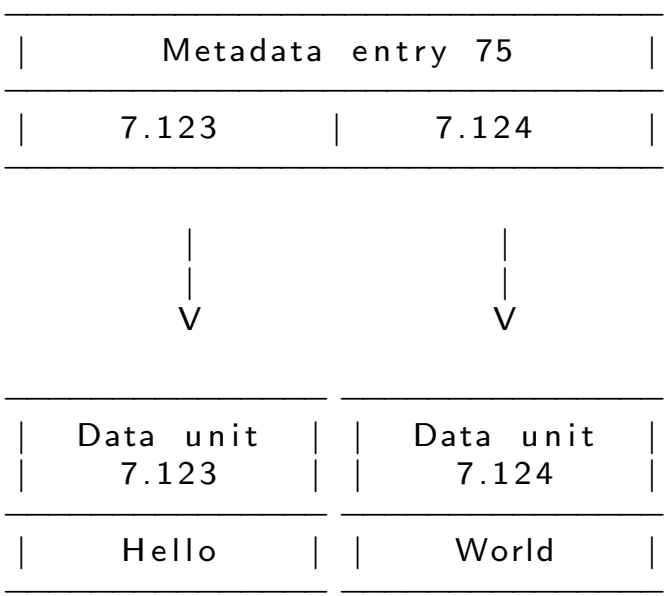
6.4 Slack space



0 = Unallocated
1 = Allocated

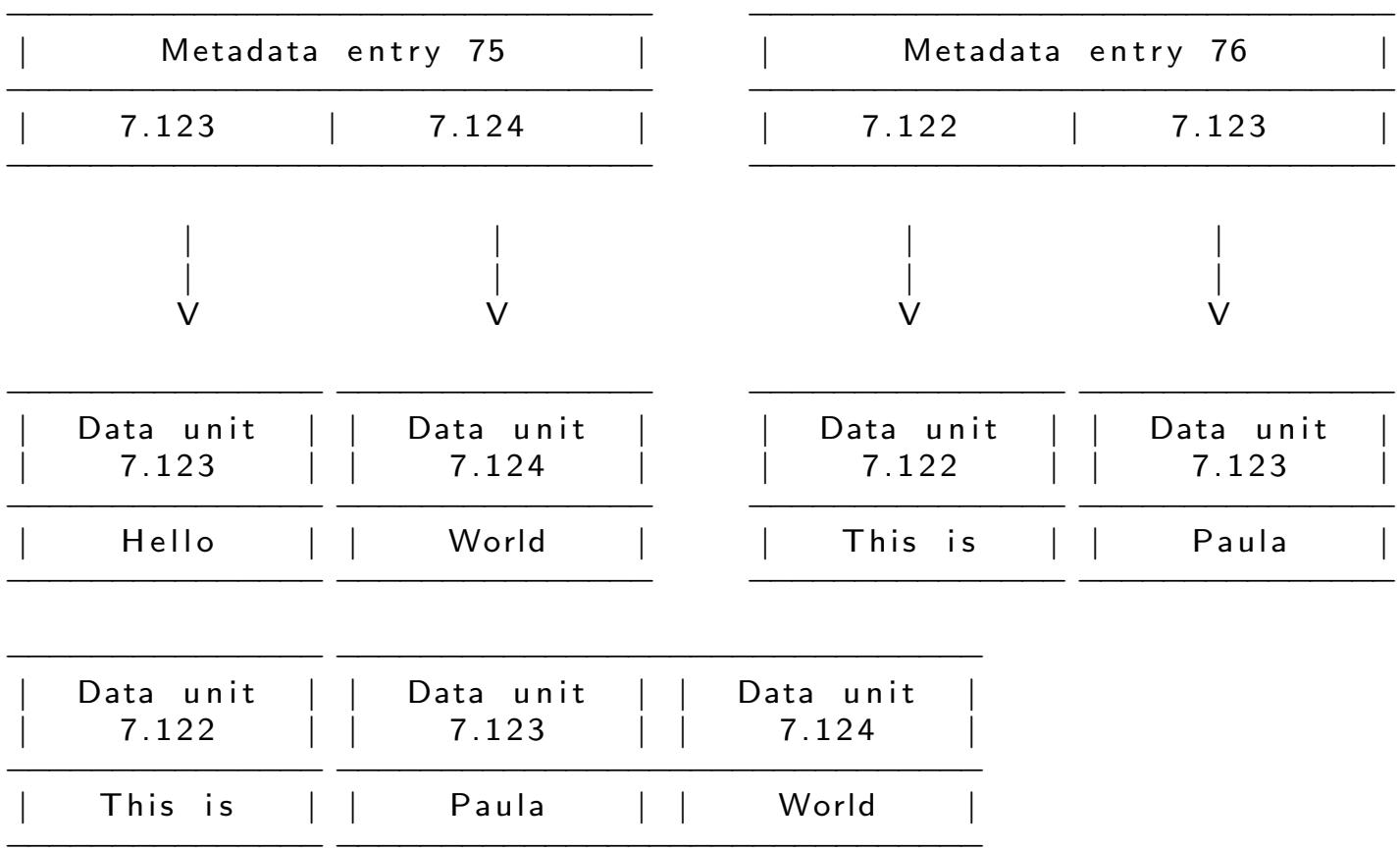
- Slack space 1 is usually wiped with zeros.
- Slack space 2 is mostly not touched.
- Maybe there are rests of deleted data.
- Complete cluster is allocated to the file

6.5 Metadata based file recovery



Exercise: Recover deleted files with Sleuthkit

6.6 File partialy over written



6.7 File System Timeline

- MAC time stamps
  - M-time: Content last modified
  - A-time: Content last accessed
  - C-time: Meta data last modified
  - B-time: File created

Exercise: Create a timealine from image `'deleted.dd'`

```
# mkdir timeline
# cd timeline

# fls -f ntfs -m d: -r ntfs.dd > fls_body
        -m d: mountpoint
        -r    recursive

# mactime -b fls_body > fls_time
```

6.7 File System Timeline

Exercise: Timeline analysis

```
Thu Jun 27 2013 12:23:08     113 ...b 35-128-1 c:/time-0001.txt
Thu Jun 27 2013 12:24:20      75 m.cb 37-128-1 c:/time-0002.txt
Thu Jun 27 2013 12:25:24      75 m.cb 38-128-1 c:/time-0003.txt
                              75 m... 41-128-1 c:/time-0003 - Copy.txt
Thu Jun 27 2013 12:26:05      75 m..b 39-128-1 c:/time-0044.txt
Thu Jun 27 2013 12:27:00      75 macb 40-128-1 c:/time-0005.txt (deleted)
Thu Jun 27 2013 12:33:50     113 m.c. 35-128-1 c:/time-0001.txt
Thu Jun 27 2013 13:07:52      75 .acb 41-128-1 c:/time-0003 - Copy.txt
Thu Jun 27 2013 13:10:36      75 ..c. 39-128-1 c:/time-0044.txt
Thu Jun 27 2013 13:14:20      20 m... 42-128-1 c:/time-0006.txt
Thu Jun 27 2013 13:56:30      20 .acb 42-128-1 c:/time-0006.txt
Thu Jun 27 2013 16:00:16     113 .a.. 35-128-1 c:/time-0001.txt
                              75 .a.. 37-128-1 c:/time-0002.txt
                              75 .a.. 38-128-1 c:/time-0003.txt
                              75 .a.. 39-128-1 c:/time-0044.txt
```

6.7 File System Timeline

```
Thu Jun 27 2013 12:23:08      113 ...b        35-128-1 c:/time-0001.txt
Thu Jun 27 2013 12:24:20       75 m.cb        37-128-1 c:/time-0002.txt
Thu Jun 27 2013 12:25:24       75 m.cb        38-128-1 c:/time-0003.txt
                               75 m...        41-128-1 c:/time-0003 - Copy.txt
Thu Jun 27 2013 12:26:05       75 m..b        39-128-1 c:/time-0044.txt
Thu Jun 27 2013 12:27:00       75 macb        40-128-1 c:/time-0005.txt (deleted)
Thu Jun 27 2013 12:33:50      113 m.c.        35-128-1 c:/time-0001.txt
Thu Jun 27 2013 13:07:52       75 .acb        41-128-1 c:/time-0003 - Copy.txt
Thu Jun 27 2013 13:10:36       75 ..c.        39-128-1 c:/time-0044.txt
Thu Jun 27 2013 13:14:20       20 m...        42-128-1 c:/time-0006.txt
Thu Jun 27 2013 13:56:30       20 .acb        42-128-1 c:/time-0006.txt
```

1. 12:23:08 time-0001.txt -> new create
2. 12:24:20 time-0002.txt -> new create
3. 12:25:24 time-0003.txt -> new create
4. 12:26:05 time-0004.txt -> new create
5. 12:27:00 time-0005.txt -> new create

6. 12:29:07 time-0001.txt -> modified content
7. 12:33:50 time-0001.txt -> 2nd modification
8. 12:29:50 time-0002.txt -> open/access file
9. 12:30:01 time-0002.txt -> close

10. 13:07:52 time-0003.txt -> copy to time-0003 - Copy.txt
11. 13:10:36 time-0004.txt -> rename to time-0044.txt

12. 13:14:20 time-0006.txt -> new created on other drive
13. 13:56:30 time-0006.txt -> copy to local drive
14. 13:58:07 time-0005.txt -> delete file



CIRCL
Computer Incident
Response Center
Luxembourg

7. Analysing files

7.1 Analysing files

- Linux commands

  `strings`
  `file`
  `exiftool`
  `md5sum`

- Online resources

  NSRL - National Software Reference Library
  VirusTotal
  CIRCL: DMA
  CIRCL: MISP Threat Sharing Platform

- Exercise: Search MD5

  `A479C4E7ED87AEDAFAD7D9936DC80115`
  `81e9036aed5502446654c8e5a1770935`



CIRCL
Computer Incident
Response Center
Luxembourg

8. String Search

8.1 String Search

- 8.1 Exercise: What about Paulas cat
  - Search for 'paula'

```
blkls -e /dev/sdc2 | srch_strings -a -td | grep -i paul
blkls -e -o 4196352 /dev/sdc | srch_strings -a -td | grep -i paul

    192158 Paul's cat is fat.........
    192185 Paul's cat is fat.........
    192212 Paul's cat is fat.........
    192239 Paul's cat is fat.........
```

  - Find back inode number

```
ifind -o 4196352 -d $((192158/4096)) /dev/sdc2

    0-128-1
```

  - Find back file name

```
ffind -o 4196352 /dev/sdc2 0-128-1

    //$MFT
```

  - Review raw content on disk

```
dd if=/dev/sdc bs=512 skip=192158 count=20 | xxd | less
dd if=/dev/sdc bs=512 skip=$((4196352 + 192158)) count=20 | xxd | less
dd if=/dev/sdc bs=512 skip=$((4196352 + 192158/512)) count=20 | xxd | less
```



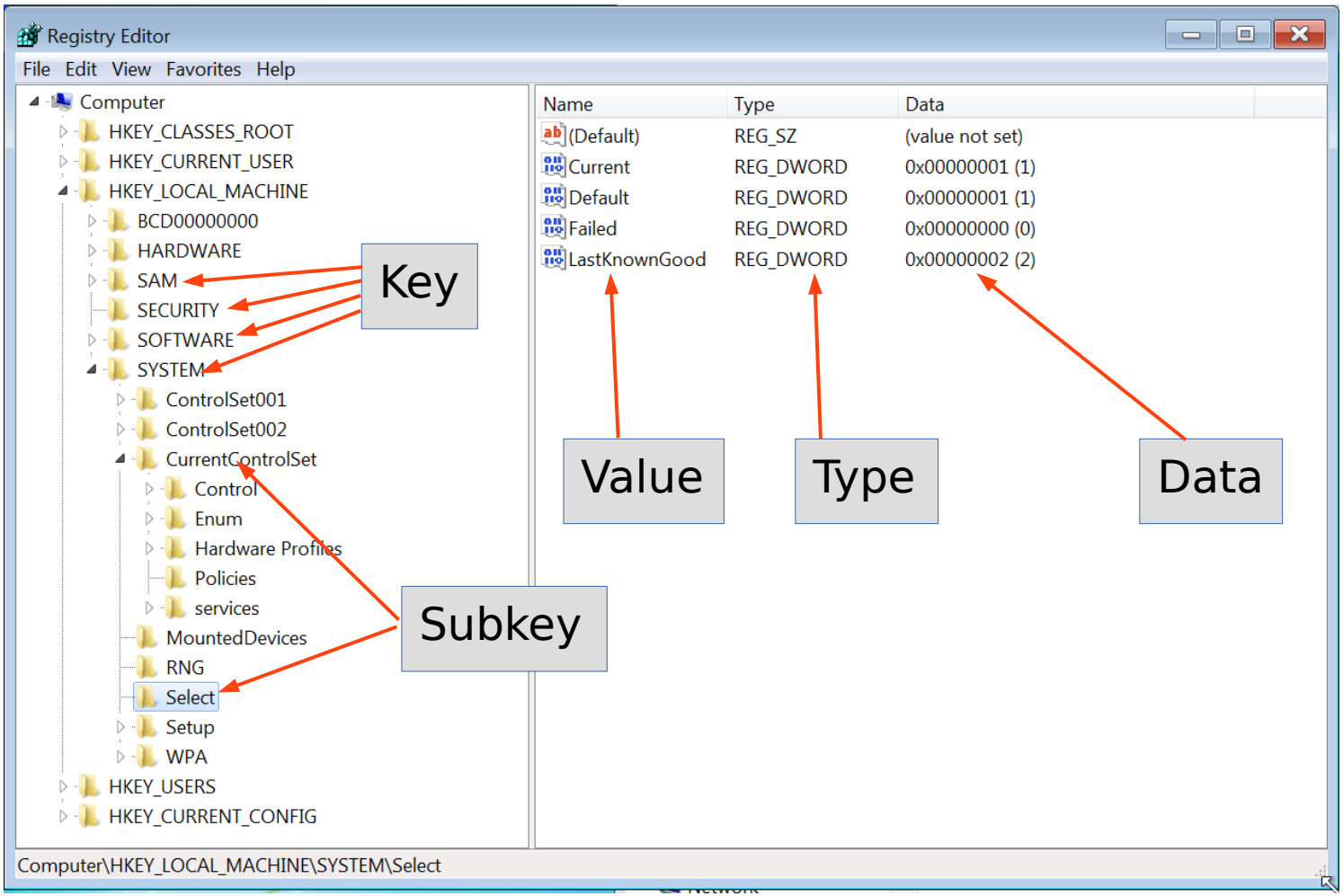
CIRCL
Computer Incident
Response Center
Luxembourg

10. Windows Registry Analysis

10.1 Windows Registry

- `https://support.microsoft.com/en-us/help/256986/`
  - A central hierarchical database
  - Contains information for operating
    - Each user
    - Installed applications
    - Existing hardware
    - Windows
  - Replaces the text-based .ini files
  - Registry hive and hive files
    - Keys, subkeys, and values
    - Windows Vista and above:
      `%SystemRoot%\system32\config`
      `SAM, SECURITY, SYSTEM, SOFTWARE`
      `%UserProfile%\NTUSER.DAT`
      `%UserProfile%\AppData\Local\Microsoft\Windows\UsrClass.dat`
- Timestamps $\rightarrow$ Timeline

10.2 Nomenclature



10.3 Example: Registry key cell

```
a0 ff ff ff  6e 6b 20 00 6f 0f 0e 3b b7 8d d1 01   ....nk .o..;....
02 00 00 00 08 5e 05 00 00 00 00 00 00 00 00 00   .....^..........
ff ff ff ff ff ff ff ff 02 00 00 00 00 21 05 00   .............!..
10 2e 00 00 ff ff ff ff 00 00 00 00 00 00 00 00   ................
14 00 00 00 10 00 00 00 00 00 00 00 0a 00 00 00   ................
49 6e 74 65 72 66 61 63 65 73 00 80 02 00 00 00   Interfaces......
```

Offsets in Bytes:			
	0	4	Size
	4	2	Node ID
	6	2	Node type
	8	8	Last write time
	...		
	76	2	Lenght of key name
	80	<76>	key name + padding

- Exercise: Calculate the size of the key cell

  `a0 ff ff ff`

- Exercise: Calculate the size of the key name

  `0a 00`

10.3 Example: Registry value cell

```
                                d8 ff ff ff  76 6b 0d 00            ....vk..
04 00 00 80 02 00 00 00 04 00 00 00 01 00 00 00  ................
4c 61 73 74 4b 6e 6f 77 6e 47 6f 6f 64 00 00 00  LastKnownGood...
```

Offsets in Bytes:			
	0	4	Size
	4	2	Node ID
	6	2	Value name length
	8	4	Data lenght
	12	4	Data offset
	16	4	value typw

- Exercise: Calculate the size of the value cell

  `d8 ff ff ff`
- Exercise: Calculate the size of the value name length

  `0d 00`

10.4 System hives

- SAM hive
  - Local users
- Security hive
  - Audit settings
  - Machine, domain SID
- System hive
  - General system configuration
  - Networking, Auto run
  - Program execution
  - USB devices
- Software hive
  - Windows version, Profiles list
  - Networking, Auto run
  - Shell extensions, Browser helper objects
  - Scheduled Tasks
  - Program execution

10.5 RegRipper

```
### Rip profiles ###
wine /usr/share/regripper/rip.exe -f sam -r caseX/registry/SAM
wine /usr/share/regripper/rip.exe -f security -r casex/registry/SECURITY
wine /usr/share/regripper/rip.exe -f system -r caseX/registry/SYSTEM
wine /usr/share/regripper/rip.exe -f software -r caseX/registry/SOFTWARE
wine /usr/share/regripper/rip.exe -f ntuser-all -r caseX/registry/NTUser.dat
wine /usr/share/regripper/rip.exe -f usrclass -r caseX/registry/UsrClass.dat

### Extract specific data ###
wine /usr/share/regripper/rip.exe -p samparse -r caseX/registry/SAM
    ...
    Username        : Locky [1000]
    Account Type    : Default Admin User
    Account Created : Tue Jul  5 14:22:17 2016 Z
    Password Hint   : c
    Last Login Date : Thu Feb  1 12:44:22 2018 Z
    Pwd Fail Date   : Wed Jan 11 09:45:59 2017 Z
    Login Count     : 36
      --> Password does not expire
      --> Password not required

wine /usr/share/regripper/rip.exe -p polacdms -r caseX/registry/SECURITY
    ...
    LastWrite Time Tue Jul  5 23:11:33 2016 (UTC)
    Machine SID: S-1-5-21-4212223026-3181619266-2879170966
    LastWrite Time Tue Jul 14 04:34:21 2009 (UTC)
    Primary Domain SID: S-1-5-
```

10.6 User hives

- Windows XP:

  `C:\Documents and Settings\<username>\NTUSER.DAT`

  `C:\Documents and Settings\<username>\Local Settings\`
  `Application Data\Microsoft\Windows\UsrClass.dat`

- Windows Vista and above:

  `C:\Users\<user>\NTUSER.DAT`

  `C:\Users\<user>\AppData\Local\Microsoft\Windows\`
  `UsrClass.dat`

10.6 User hives: NTUSER.DAT

- AutoStart
  - Run Key
    - `\Software\Microsoft\Windows\CurrentVersion\Run\`
    - Executed at user login
    - Provide persistence *malware*
    - No admin privileges required
  - RunOnce Key
    - `\Software\Microsoft\Windows\CurrentVersion\RunOnce\`
    - Values will be launched and deleted
  - Legacy and other AutoStart
    - `\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run\`
    - `\Software\Microsoft\Windows NT\CurrentVersion\Windows\'load' and 'run' values`

10.6 User hives: NTUSER.DAT

- Program execution
  - UserAssist
    - `\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\`
    - Windows Explorer Shell users interaction
    - Subkey values: ROT-13
    - Subkey values: Path, Count, Last action
  - Application Compatibility Assistant

- Other activities
  - Typed paths in Windows Exporer Shell
  - Typed URLs
  - Searches
  - File associations

  ...

10.6 User hives: NTUSER.DAT

- File access
  - RecentDocs

    Example: '.png' files

```
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.png
LastWrite Time Fri Jan 12 15:00:52 2018 (UTC)
MRUListEx = 3,2,0,1
  3 = photo-123.png
  2 = paint.png
  0 = face.png
  1 = flower.png
```

  - Common Dialogs

    Example: 'Open' and 'Save As...'

```
OpenSavePidlMRU\exe
LastWrite Time: Tue Jul  5 14:40:46 2016
Note: All value names are listed in MRUListEx order.

  Users\avast_free_antivirus_setup_online.exe
  Users\Thunderbird Setup 45.1.1.exe
  Users\Firefox Setup Stub 47.0.1.exe
```

10.6 User hives: UsrClass.dat

- Program execution: By 'cmd.exe'
  - MUI Cache
    - XP
      `Software\Microsoft\Windows\ShellNoRoam\MUICache\`
    - Vista
      `Local Settings\Software\Microsoft\Windows\Shell\MUICache`

- File access
  - Shell Bags
    - MRU time
    - File system time stamps
    - ZIP subfolders
    - FTP access by Windows Explorer
    - MFT reference
    - Path



CIRCL
Computer Incident
Response Center
Luxembourg

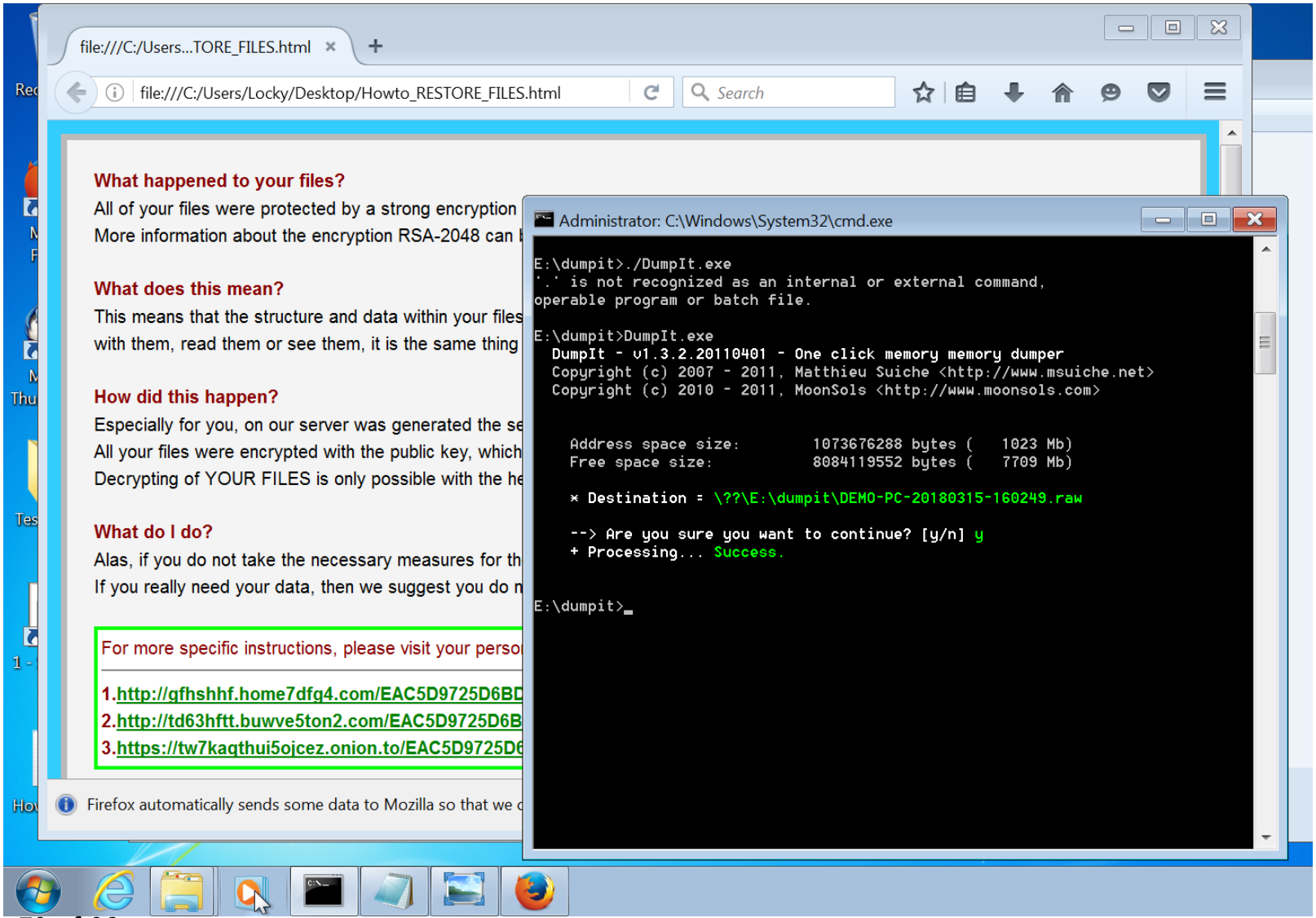
11. Memory Forensics

11.1 Memory Forensics

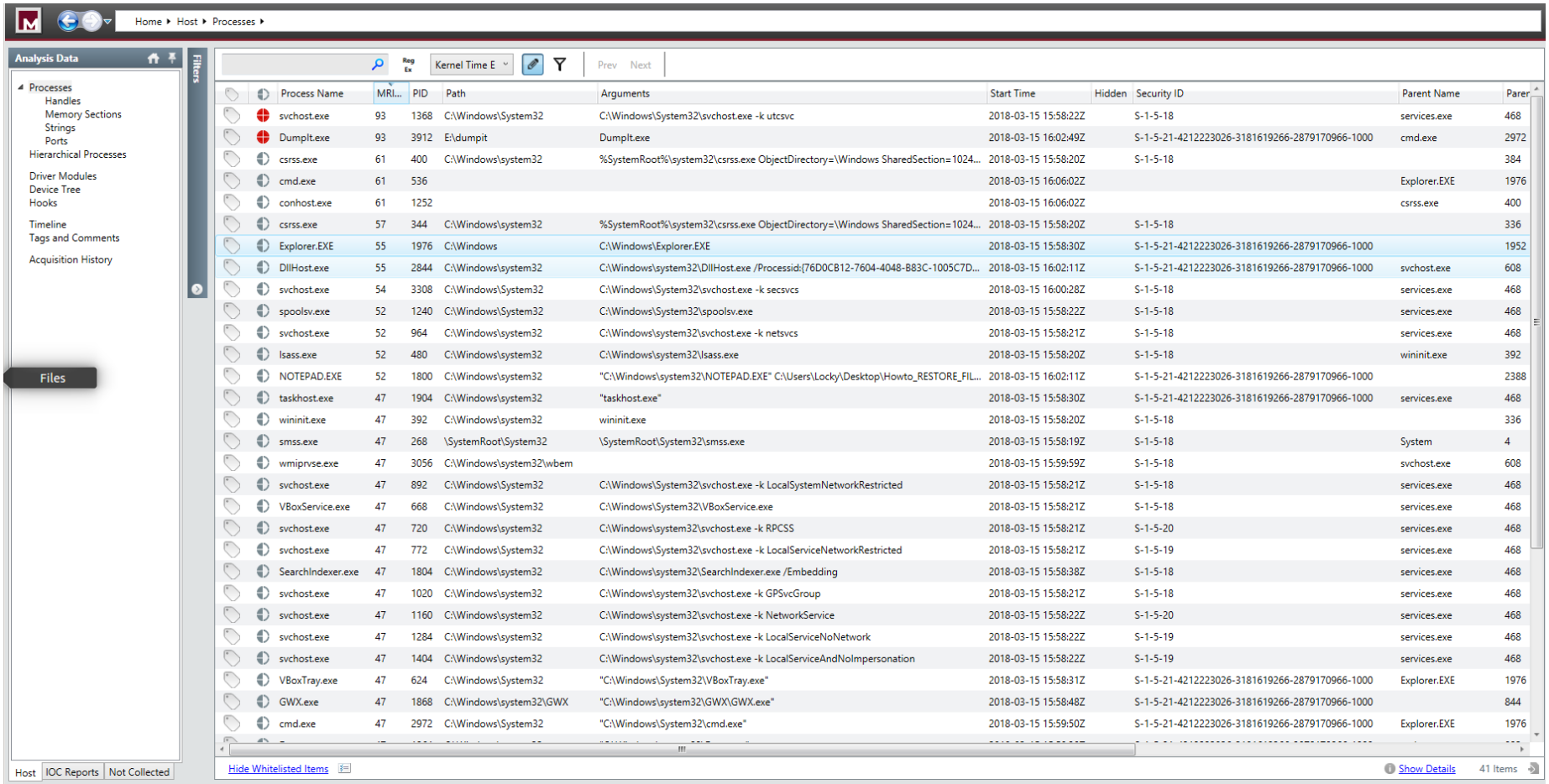
- Find memory data
  - Hibernation file: `hiberfil.sys`
  - Page file, swap area: `pagefile.sys`
  - Memory dump

- Information expected
  - Processes - hidden
  - Processes - analysis
  - Network - services
  - Network - connections
  - DLL - analysis
  - Registry - analysis
  - Passwords - clear text
  - Malware

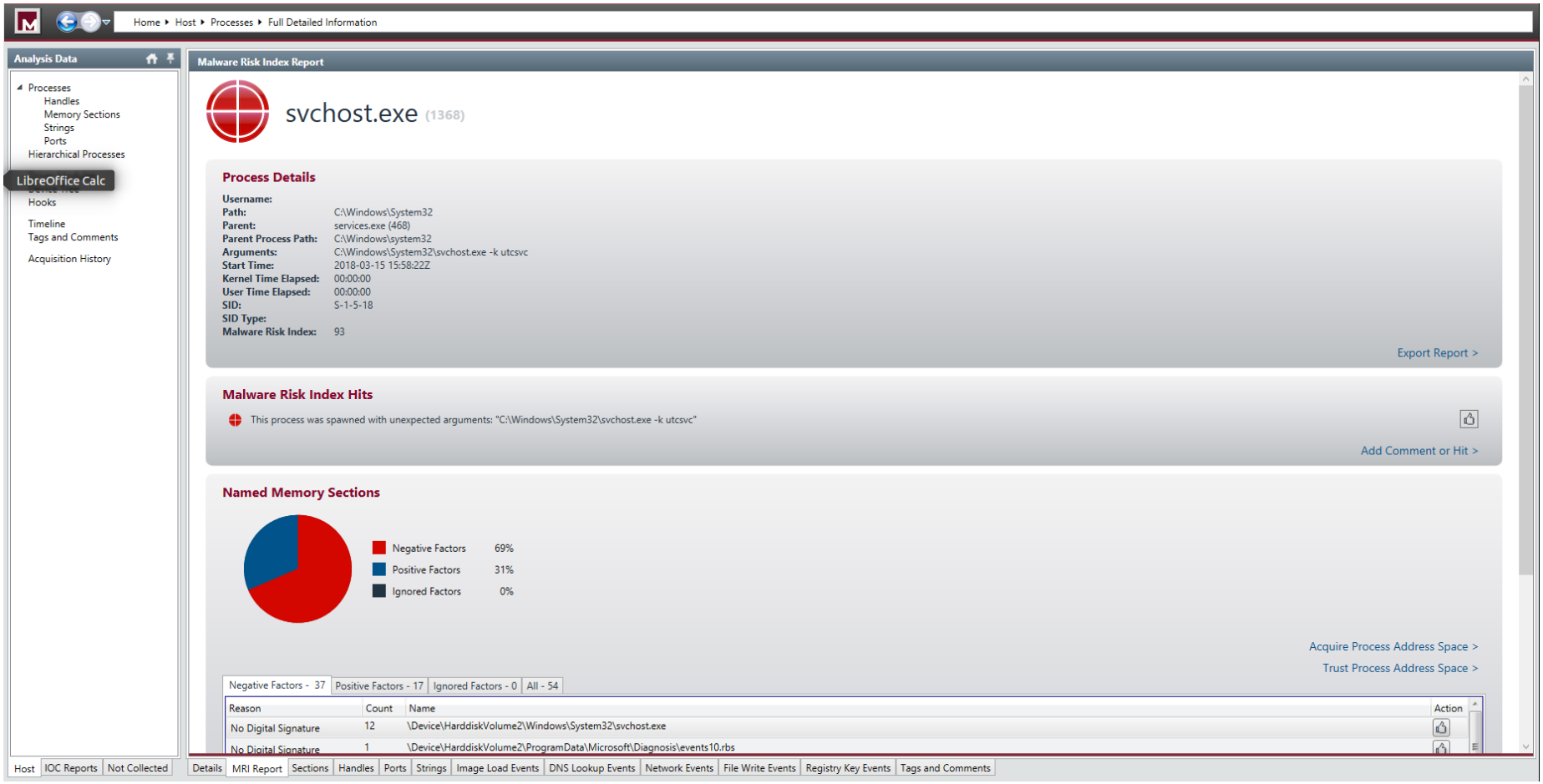
11.1 DumpIt


file:///C:/Users...TORE_FILES.html
file:///C:/Users/Locky/Desktop/Howto_RESTORE_FILES.html
Search
What happened to your files?
All of your files were protected by a strong encryption
More information about the encryption RSA-2048 can
What does this mean?
This means that the structure and data within your files
with them, read them or see them, it is the same thing
How did this happen?
Especially for you, on our server was generated the se
All your files were encrypted with the public key, which
Decrypting of YOUR FILES is only possible with the he
What do I do?
Alas, if you do not take the necessary measures for th
If you really need your data, then we suggest you do n
For more specific instructions, please visit your perso
1.http://gfhshhf.home7dfg4.com/EAC5D9725D6BD
2.http://td63hftt.buwve5ton2.com/EAC5D9725D6B
3.https://tw7kaqthui5ojcez.onion.to/EAC5D9725D6
Firefox automatically sends some data to Mozilla so that we
Administrator: C:\Windows\System32\cmd.exe
E:\dumpit>./DumpIt.exe
'.' is not recognized as an internal or external command,
operable program or batch file.
E:\dumpit>DumpIt.exe
DumpIt - v1.3.2.20110401 - One click memory memory dumper
Copyright (c) 2007 - 2011, Matthieu Suiche <http://www.msuiche.net>
Copyright (c) 2010 - 2011, MoonSols <http://www.moonsols.com>
Address space size: 1073676288 bytes (1023 Mb)
Free space size: 8084119552 bytes (7709 Mb)
* Destination = \??\E:\dumpit\DEMO-PC-20180315-160249.raw
--> Are you sure you want to continue? [y/n] y
+ Processing... Success.
E:\dumpit>

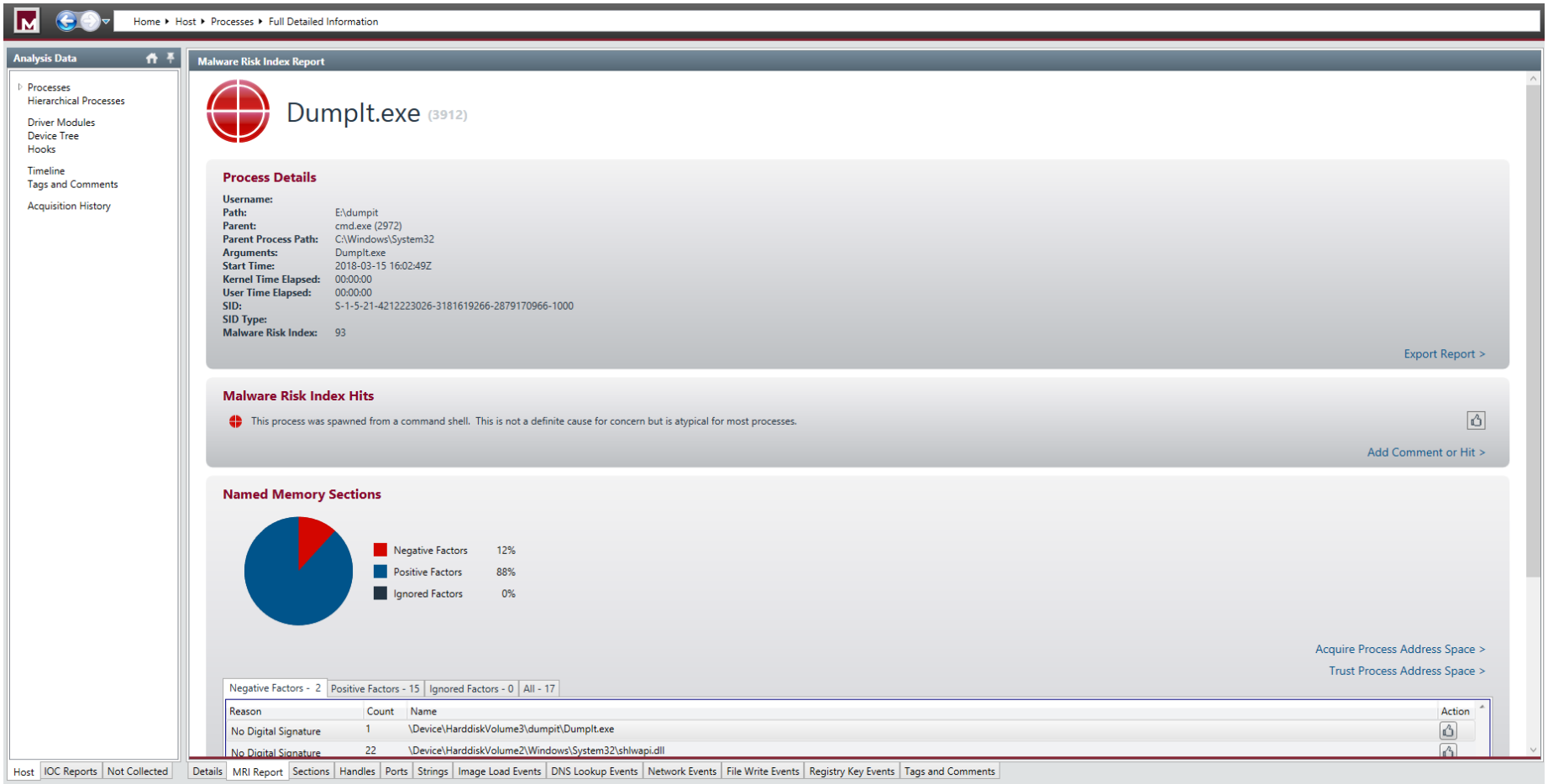
11.2 Mandiant Redline - Malware Risk Index



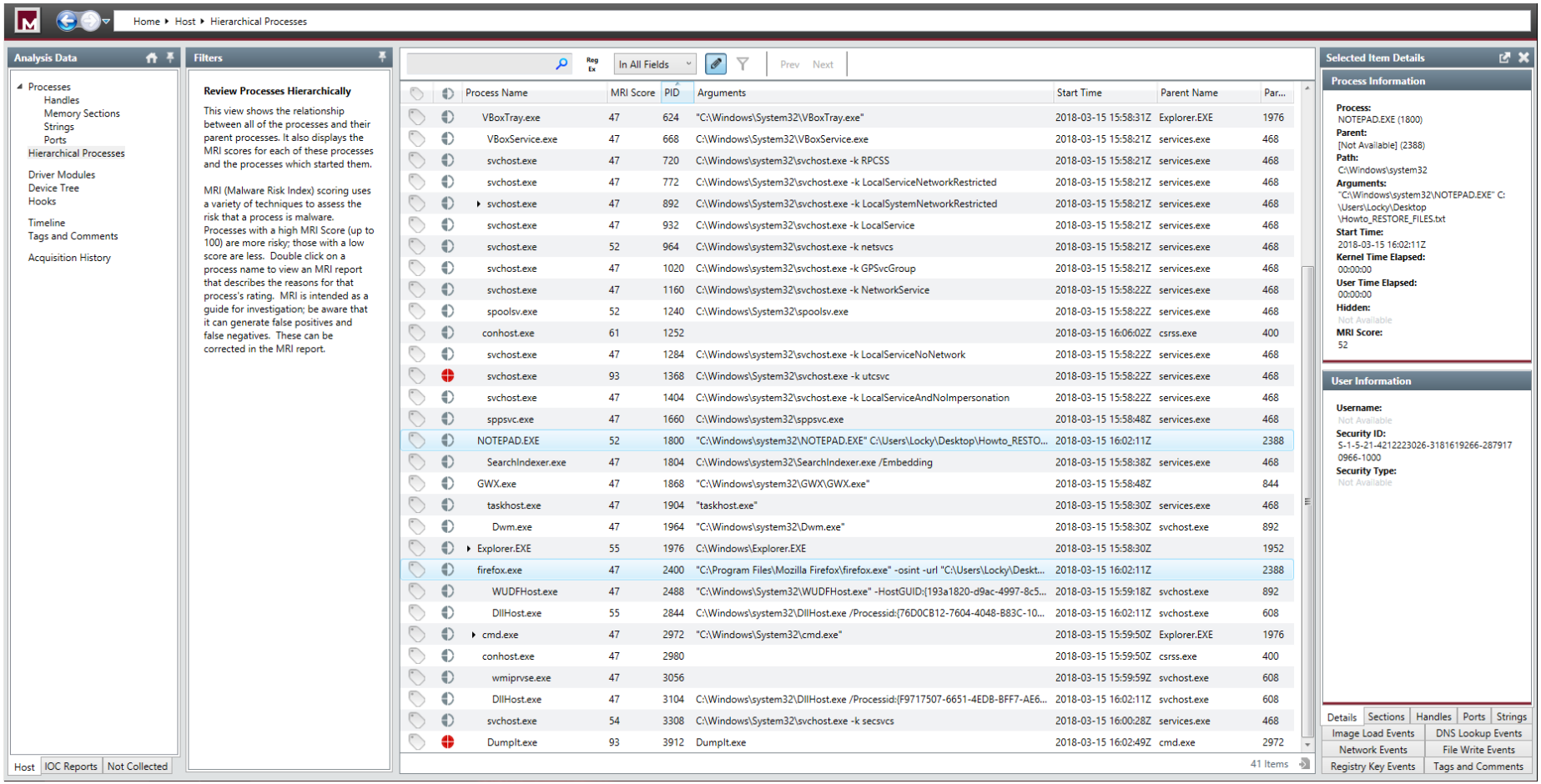
11.2 Mandiant Redline - Malware Risk Index



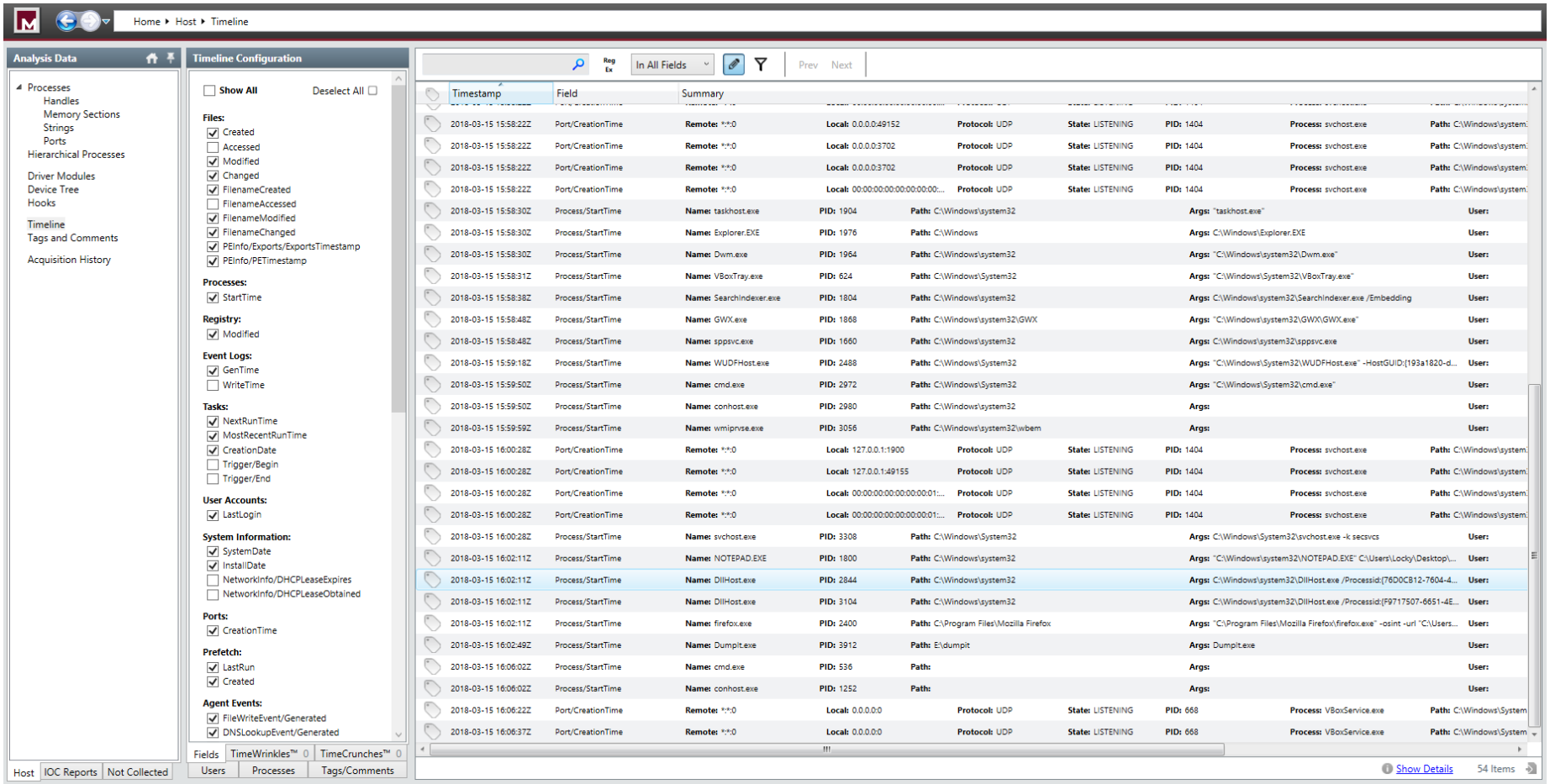
11.2 Mandiant Redline - Malware Risk Index



11.2 Mandiant Redline - Hierarchical



11.2 Mandiant Redline - Timeline



11.3 Volatility: Overview

```
volatility -h

...
imagecopy       Copies a physical address space out as a raw DD image
imageinfo       Identify information for the image
...
pslist          Print all running processes by following the EPROCESS lists
psscan          Scan Physical memory for _EPROCESS pool allocations
pstree          Print process list as a tree
psxview         Find hidden processes with various process listings
...
sockets         Print list of open sockets
sockscan        Scan Physical memory for _ADDRESS_OBJECT objects (tcp sockets)
...
```

```
volatility -f [filename] [plugin] [options]
```

```
volatility -f DEMO-PC-20180315.raw imageinfo
```

11.3 Volatility: Overview

```
volatility -f DEMO-PC-20180315.raw imageinfo
```

```
Volatility Foundation Volatility Framework 2.3.1
Determining profile based on KDBG search...

          Suggested Profile(s) : Win7SP0x86, Win7SP1x86
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (memory/DEMO-PC-20180315-160249.r
                      PAE type : No PAE
                           DTB : 0x185000L
                          KDBG : 0x82954c70
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x82955d00
             KUSER_SHARED_DATA : 0xffdf0000
           Image date and time : 2018-03-15 16:02:54 UTC+0000
     Image local date and time : 2018-03-15 17:02:54 +0100
```

```
volatility --profile=Win7SP1x86 -f [filename] [plugin]
[options]
```

11.3 Volatility: Process Analysis

`pslist`

- Running processes
- Process IP - PID
- Parent PIP - PPID
- Start time

`pstree`

- Like `pslist`
- Visual child-parent relation

`psscan`

- Find hidden processes
- Find inactive processes

`psxview`

- Run and compare some tests
- Correlate `psscan` and `pslist`

11.3 Volatility: Process Analysis

```
volatility --profile=Win7SP1x86 -f DEMO-PC-20180315.raw
pslist
```

Offset(V)	Name	PID	PPID	Thds	Hnds	Ses	Wow64	Start
0x84233a00	System	4	0	80	384	----	0	2018-03-15 15:58:19
0x85a40d28	smss.exe	268	4	2	29	----	0	2018-03-15 15:58:19
0x8534eb78	csrss.exe	344	336	9	359	0	0	2018-03-15 15:58:20
0x842a69d8	wininit.exe	392	336	3	75	0	0	2018-03-15 15:58:20
0x86777b18	csrss.exe	400	384	7	301	1	0	2018-03-15 15:58:20
0x8529b030	winlogon.exe	448	384	3	111	1	0	2018-03-15 15:58:20
0x852c9030	services.exe	468	392	8	188	0	0	2018-03-15 15:58:20
0x85326030	lsass.exe	480	392	7	478	0	0	2018-03-15 15:58:20
0x8536c030	lsm.exe	488	392	10	145	0	0	2018-03-15 15:58:20
...								
...								
0x85234950	sppsvc.exe	1660	468	6	164	0	0	2018-03-15 15:58:48
0x8435d508	WUDFHost.exe	2488	892	8	205	0	0	2018-03-15 15:59:18
0x857a92e0	cmd.exe	2972	1976	1	21	1	0	2018-03-15 15:59:50
0x852960f0	conhost.exe	2980	400	2	54	1	0	2018-03-15 15:59:50
0x85a93778	WmiPrvSE.exe	3056	608	6	114	0	0	2018-03-15 15:59:59
0x8439b030	svchost.exe	3308	468	12	342	0	0	2018-03-15 16:00:28
0x844f03e8	notepad.exe	1800	2388	2	77	1	0	2018-03-15 16:02:11
0x845f9708	firefox.exe	2400	2388	44	516	1	0	2018-03-15 16:02:11
0x84558a18	dllhost.exe	2844	608	8	238	1	0	2018-03-15 16:02:11
0x84558d28	dllhost.exe	3104	608	4	76	1	0	2018-03-15 16:02:11
0x8460b030	DumpIt.exe	3912	2972	2	39	1	0	2018-03-15 16:02:49

11.4 Volatility: Network Analysis

`connections`

- Windows XP and 2003 Server
- Active connections

`connscan`

- Windows XP and 2003 Server
- Terminated connections

`sockets`

- Listening sockets
- Additional networking data

11.4 Volatility: Exercise

```
vol --profile=WinXPSP3x86 -f cridex.vmem connections
```

Offset(V)	Local Address	Remote Address	Pid
0x81e87620	172.16.112.128:1038	41.168.5.140:8080	1484

```
vol --profile=WinXPSP3x86 -f cridex.vmem connscan
```

Offset(P)	Local Address	Remote Address	Pid
0x02087620	172.16.112.128:1038	41.168.5.140:8080	1484
0x023a8008	172.16.112.128:1037	125.19.103.198:8080	1484

```
vol --profile=WinXPSP3x86 -f cridex.vmem pslist
```

Offset(V)	Name	PID	PPID	Thds	Hnds	Start
...						
0x821dfda0	svchost.exe	1056	652	5	60	2012-07-22 02:42:3
0x82295650	svchost.exe	1220	652	15	197	2012-07-22 02:42:3
0x821dea70	explorer.exe	1484	1464	17	415	2012-07-22 02:42:3
0x81eb17b8	spoolsv.exe	1512	652	14	113	2012-07-22 02:42:3
0x81e7bda0	reader_sl.exe	1640	1484	5	39	2012-07-22 02:42:3
0x820e8da0	alg.exe	788	652	7	104	2012-07-22 02:43:0

11.5 Volatility: Other plugins

- DLL analysis
- Registry analysis
- Timeline

```
vol --profile=WinXPSP3x86 -f cridex.vmem getsids
```

```
...
explorer.exe (1484): S-1-5-21-789336058-261478967-1417001333-1003
...
```

```
vol --profile=WinXPSP3x86 -f cridex.vmem malfind
```

```
Process: explorer.exe Pid: 1484 Address: 0x1460000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 33, MemCommit: 1, PrivateMemory: 1, Protection: 6
0x01460000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x01460010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......


Process: reader_sl.exe Pid: 1640 Address: 0x3d0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 33, MemCommit: 1, PrivateMemory: 1, Protection: 6
0x003d0000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x003d0010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
```



CIRCL
Computer Incident
Response Center
Luxembourg

12. Outlook

12.1 Training materials

- Digital forensics testing images

  `http://dftt.sourceforge.net/`
  `http://digitalcorpora.org/corpora/disk-images`

- Memory forensics

  `https://github.com/volatilityfoundation/volatility/wiki/Memory-Samples`

- Network forensics: PCAP files

  `http://wiki.xplico.org/doku.php?id=pcap:pcap`
  `https://wiki.wireshark.org/SampleCaptures`

12.2 Books

- Digital Forensics with Kali Linux

  Shiva V.N. Parasram
  Packt Publishing
  ISBN-13: 978-1-78862-500-5

- Practical Forensic Imaging

  Bruce Nikkel
  No Starch Press
  ISBN-13: 978-1-59-327793-2

- Digital Forensics with Open Source Tools

  Cory Altheide, Harlan Carvey
  Syngress
  ISBN-13: 978-1-59-749586-8

12.2 Books

- File System Forensic Analysis

  Brian Carrier
  Pearson Education
  ISBN-13: 978-0-32-126817-4

- Windows Registry Forensics

  Harlan Carvey
  Syngress, 2nd
  ISBN-13: 978-0-12-803291-6

- Forensic Computing: A Practitioner's Guide

  Anthony Sammes, Brian Jenkinson
  Springer
  ISBN-13: 978-1-85-233299-0

12.3 Further topics:

LNK files

Prefetch files

Volume shadow copy & Restore points