Computer autopsies

How to conduct forensic expertise using open source tools and develop Autopsy plugins

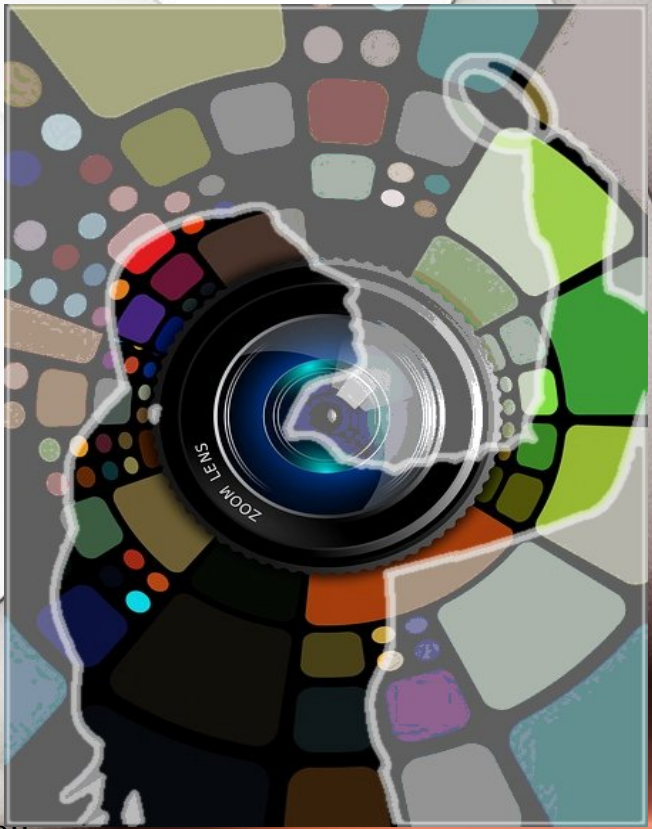
Ing María Andrea Vignau
Forensic Criminal Expert
Chaco's Judicial Branch

Computer autopsies

1) Get evidence
2) Make forensic copies
3) Data analysis using Autopsy
4) Extending Autopsy with Python
5) My example plugin.

Get evidence

- Identify devices with storage capacity
- Photograph
- See if they are turned on or off
- Evaluate RAM capture



Get evidence

- Seizure
  - The usual procedure
- Partial forensic copy on the site
  - Special servers

Get evidence

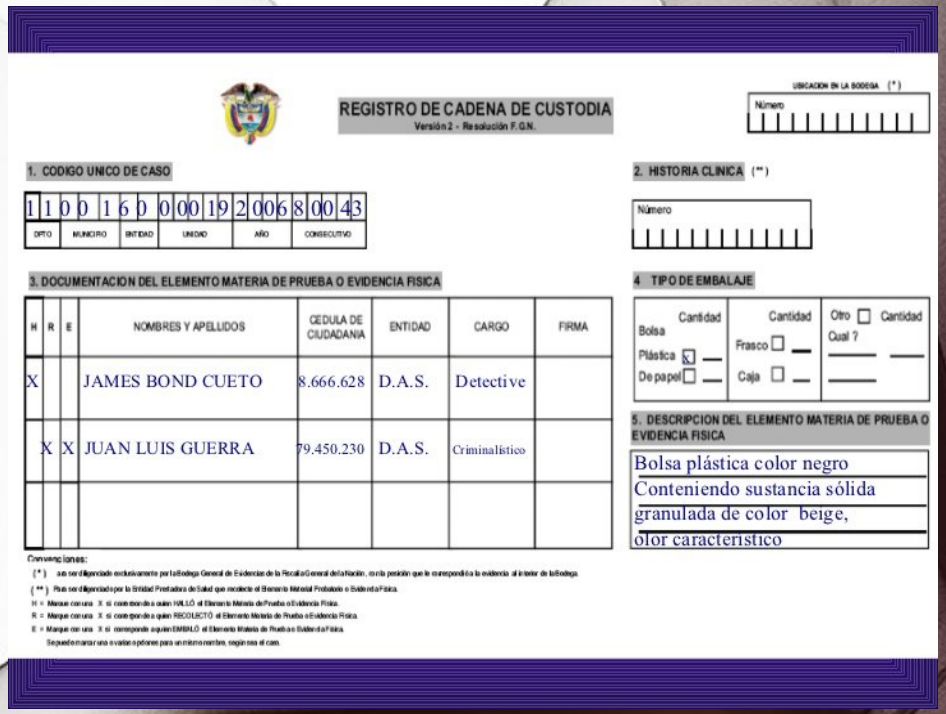
Preservation

- **Avoid** bumps, moisture
- **Wrap** preventing access to ports, or disarmament,
- **Sign** wrapping paper.

Get evidence

Chain of custody

It contains **every person** who was responsible for the integrity of the evidence.



REGISTRO DE CADENA DE CUSTODIA
Versión 2 - Resolución F.G.N.

UBICACION EN LA BODEGA (*)
Número

1. CODIGO UNICO DE CASO

1	1	0	0	1	6	0	0	0	0	1	9	2	0	0	6	8	0	0	4	3
DPTO		MUNICIPIO			ENTIDAD		UNIDAD				AÑO				CONSECUTIVO					

2. HISTORIA CLINICA (**)
Número

3. DOCUMENTACION DEL ELEMENTO MATERIA DE PRUEBA O EVIDENCIA FISICA

H	R	E	NOMBRES Y APELLIDOS	CEDULA DE CIUDADANIA	ENTIDAD	CARGO	FIRMA
X			JAMES BOND CUETO	8.666.628	D.A.S.	Detective	
	X	X	JUAN LUIS GUERRA	79.450.230	D.A.S.	Criminalístico	

4. TIPO DE EMBALAJE

Bolsa	Cantidad	Frasco	Cantidad	Otro	Cantidad
Plástica X		Caja		Cual ?	
De papel					

5. DESCRIPCION DEL ELEMENTO MATERIA DE PRUEBA O EVIDENCIA FISICA

Bolsa plástica color negro
Conteniendo sustancia sólida
granulada de color beige,
olor característico

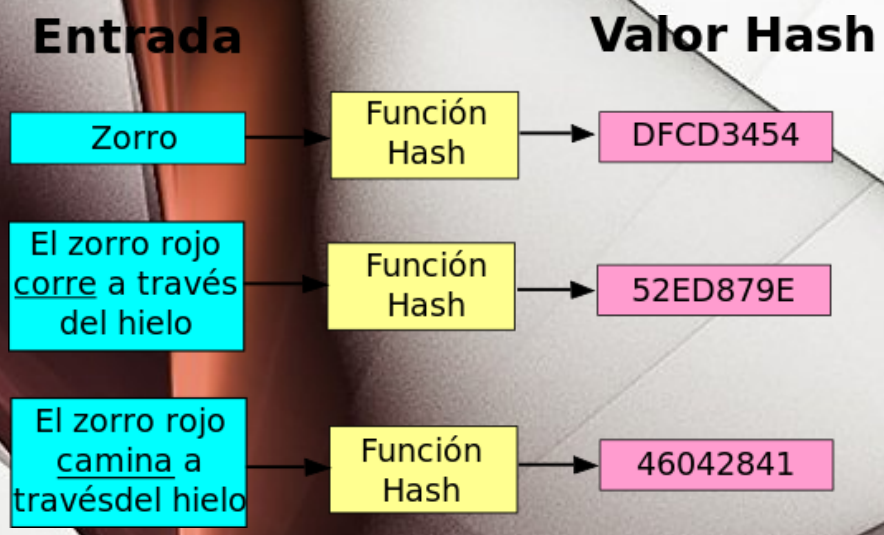
Convenciones:
(*) ... ser diligenciado exclusivamente por la Bodega General de Evidencias de la Fiscalía General de la Nación, en la posición que le corresponde a la evidencia al interior de la Bodega.
(**) Puede ser diligenciado por la Entidad Prestadora de Salud que recolecta el Elemento Material Probatorio o Evidencia Física.
H = Marque con una X si corresponde a quien HALLÓ el Elemento Material de Prueba o Evidencia Física.
R = Marque con una X si corresponde a quien RECOLECTÓ el Elemento Material de Prueba o Evidencia Física.
E = Marque con una X si corresponde a quien EMBALÓ el Elemento Material de Prueba o Evidencia Física.
Se puede marcar una o varias o personas para un mismo nombre, según sea el caso.

Make forensic copies

- Create a **forensic copy**, bit by bit, including non allocated areas.
- **Risks** that might result from direct evidence process are **avoided**.

Make forensic copies

Ensures the integrity of the data, and the evidence.



Hash function:

Input: a set of elements, strings

Output: in finite range

Projection of the set U on the set M

Make forensic copies

Removing the hard disk.

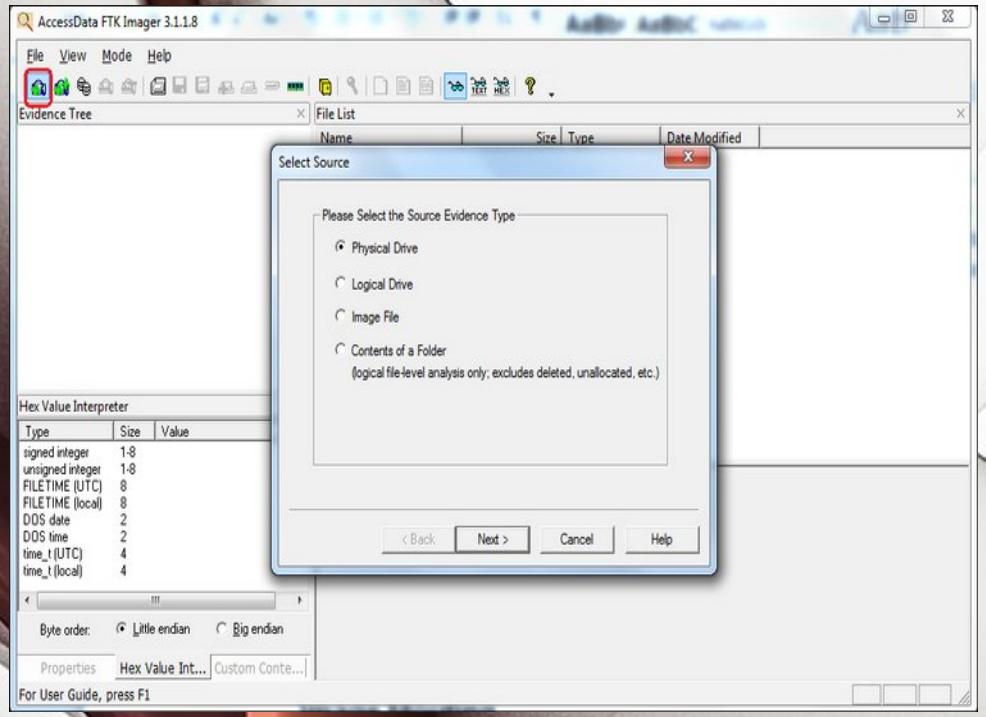
- Use forensic duplicator
  - Tableau TD3
- Connect to another PC, with USB adapter
  - Configure read-only
  - Use Hardware or Software

Make forensic copies

Windows

- Leave the USB ports read-only
- **USB Disk Access Manager.**

Make forensic copies



Windows

- **Freeware**
  - **FTK IMAGER**
- Use forensic file format
  - **Expert witness format**

Make forensic copies

Without removing the hard disk:

- **Boot** from optical disk or pendrive.
- Use a specialized distribution.
- Use forensic mode
- **read-only**

Make forensic copies

Discover devices

- `Fdisk -l`

Mount in read-only mode

- `sudo mkdir /media/2tb`
- `sudo mount -o ro /dev/sda1 /media/2tb`

Make forensic copies

Linux

- Easy to use software
  - **GuyMager**
- Use forensic file format
  - **Expert witness format**

Evidence Analysis

Autopsy SleuthKit v4.8

- Open Source
- Extensible
- Mature (v1.0 year 2001)
- Multiplataform
- Multiuser

Evidence Analysis

- Create a new case
- Add evidence
- Automatic analysis
- Manual analysis
- Reports

Evidence Analysis

- **Create** a new case
  - Enter basic data
- **Add** evidence
  - **Forensic Image**
  - Device
  - Others

Evidence Analysis

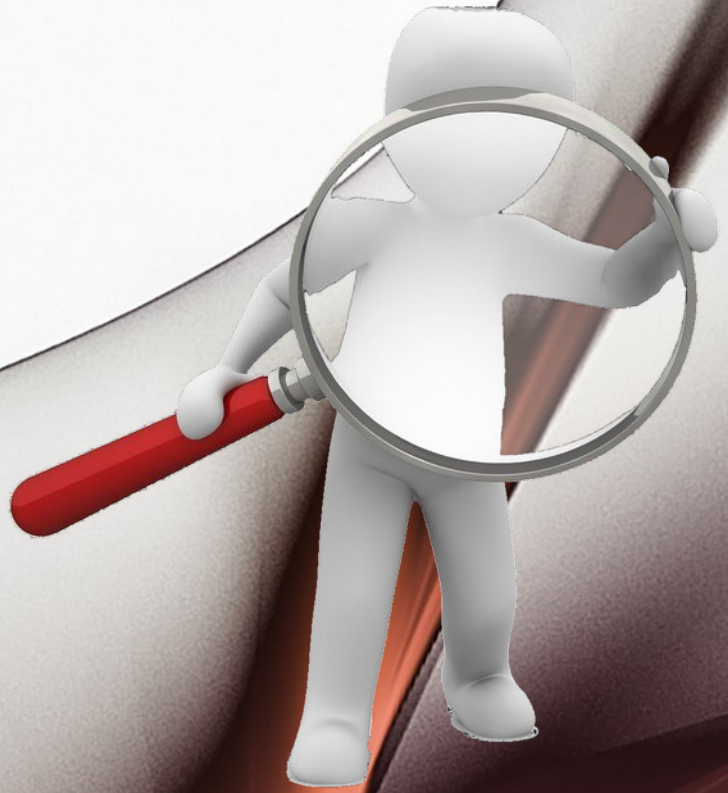
- File systems
- Forensic Images
- Compressed files
- Virtual machines
- Carving

Data sources

Evidence Analysis

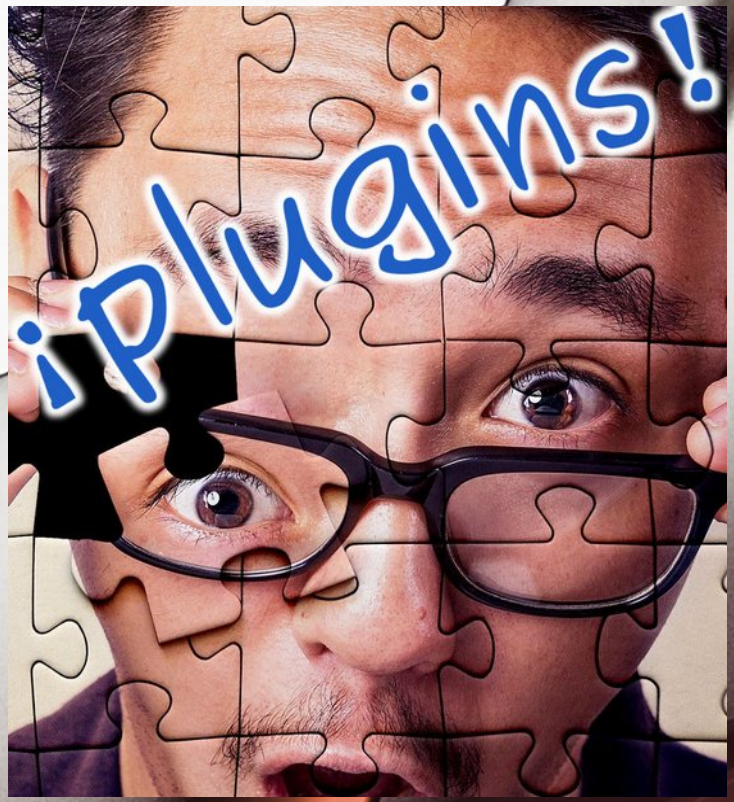
Process

- Select **plugins**
- Each one specially reviews the evidence



Evidence Analysis

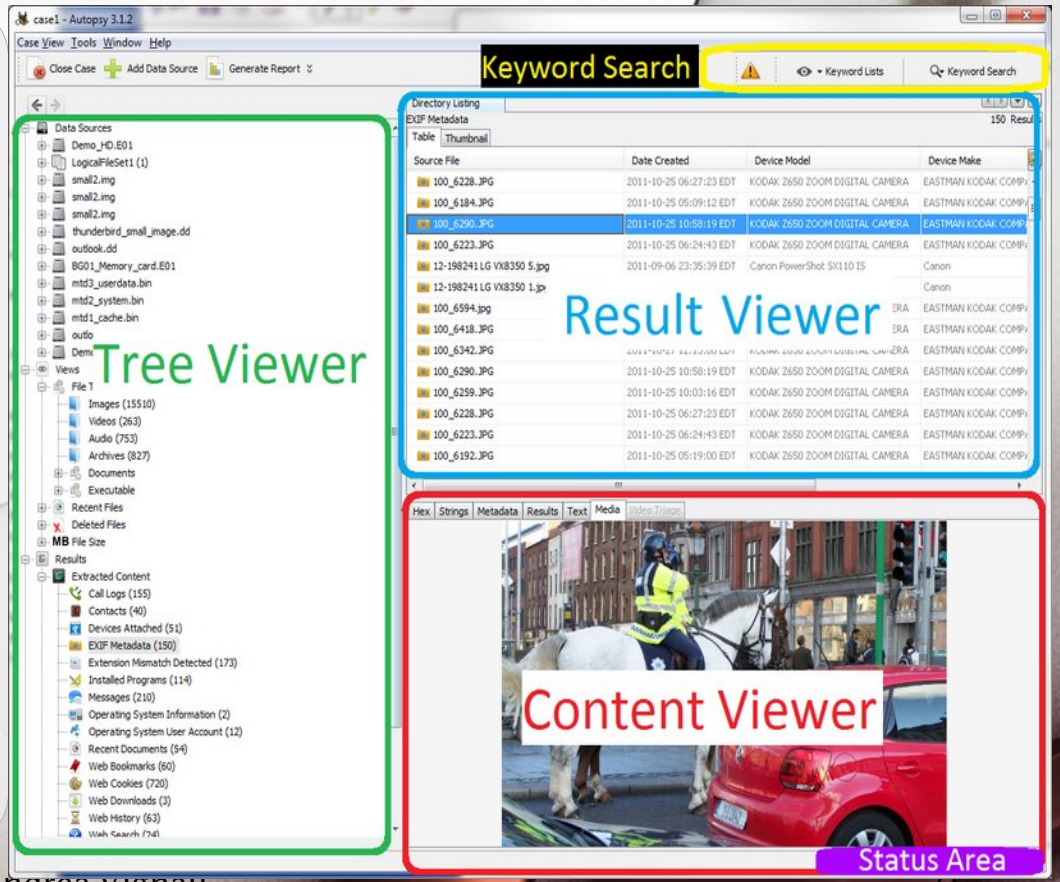
- Hash the evidence
- MIME types are identified by their file signature
- Unzip files
- They extract images embedded in documents
- Parse browser data



Evidence Analysis

Manual analysis

- **Check** manually
- **Label** the elements according to the object of the investigation

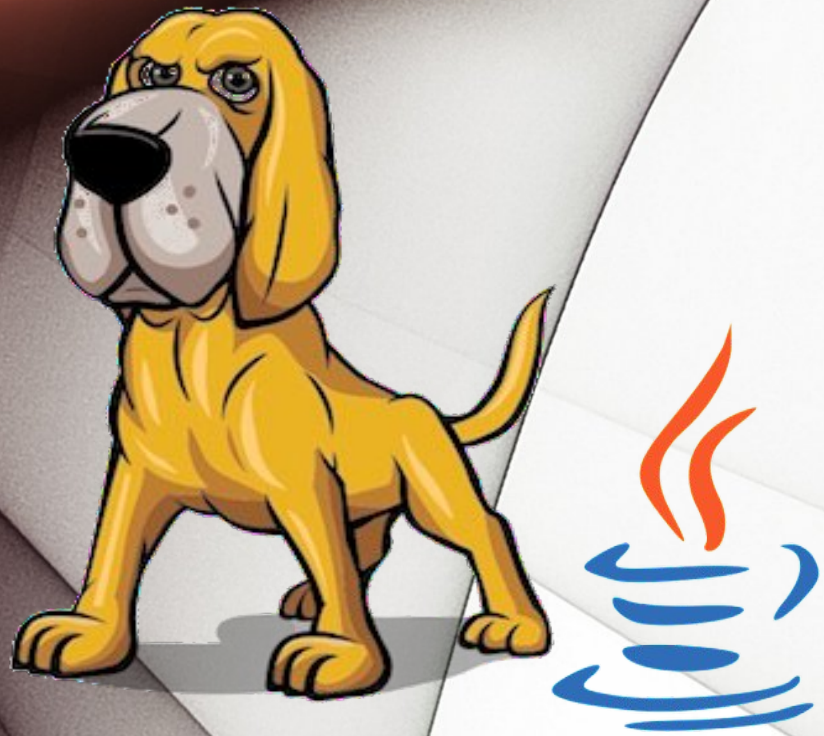


Reports

Reports

- Ready to use
- Extendable
- Many output formats.
  - HTML, PDF, etc.

Extending Autopsy with Python



- **SleuthKit** Framework
- Developed in **JAVA**
- Extensible using
  - Java
  - **Python**

Extending Autopsy with Python

Python implementations

- **Cpython:** Coded in C.
- **Jython:** Coded in Java
- PyPy
- IronPython … etc

Extending Autopsy with Python

- Configure IDE
- Create skeleton
- Choose module type
- Choose output format
- Copy and adapt tutorial model.

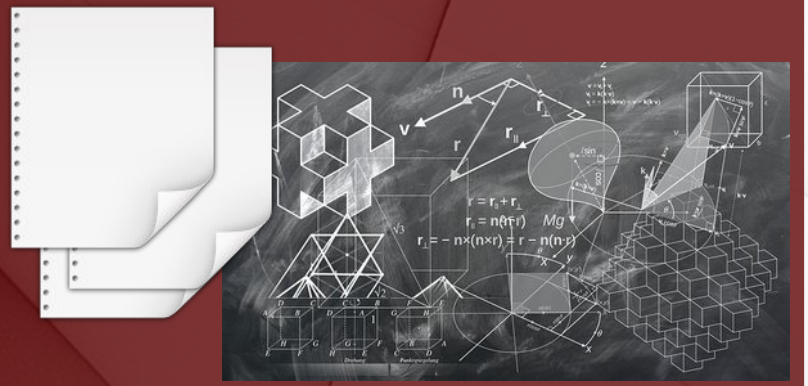
Extending Autopsy with Python

I ntegrated
D evelopment
E nvironments

- IntelliJ IDEA
- NetBeans

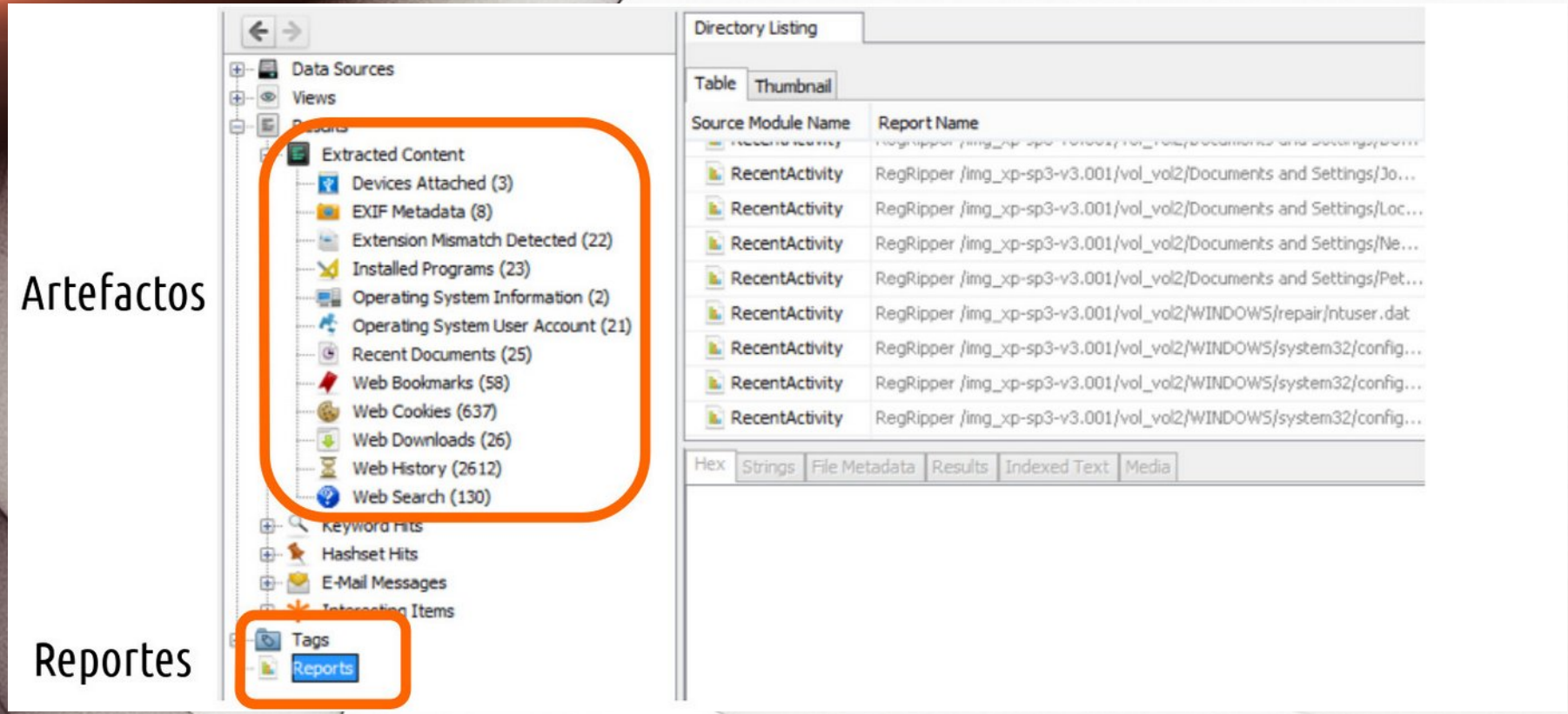
Extending Autopsy with Python

Outputs



- **Report**: easiest
- Artifacts on the board.
  - Type
  - Associated file
  - Attributes: pairs of
    - Name, Value

Extending Autopsy with Python



Extending Autopsy with Python

File Processing Module

Receives and analyzes each file's content in data sources added to case.

Extending Autopsy with Python

Data Source Process Module

- Used if you know where the file'll be.
- With external tools
- Refers to an **entire** data source.

Extending Autopsy with Python

Report Module

- Runs after the analysis to create a report output.
- Can be used data from files, artifacts and tagged by user
- HTML, XML, CVS

Extending Autopsy with Python

What kind of module suits me?

- Should I go through each file?
- Do I know exactly what file I'm looking for?
- Should I run it at the end, after manual analysis?

My example plugin

- File's hash is precisely identified.
- Database is in NIST's NSRL for Autopsy.

Well-known files

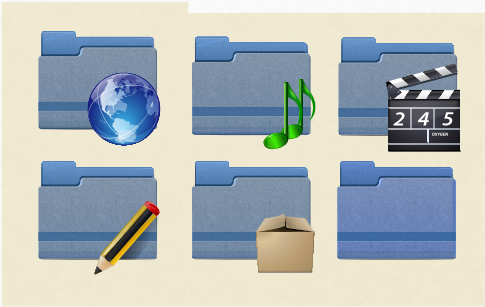
NIST National Institute of Standards and Technology U.S. Department of Commerce

My example plugin

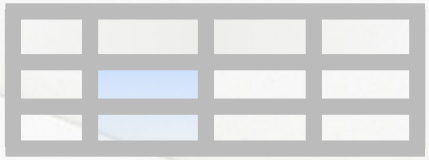
File

Well known Hash

Copies file content clasified by MIME type



Appends file's metadata to CSV file



My example plugin

- Generate the class of report inherited from **`GeneralReportModuleAdapter`**
- Set name, description and path properties to the output file

```
from org.sleuthkit.autopsy.report \
    import GeneralReportModuleAdapter

class NotKnownBackup(
    GeneralReportModuleAdapter):

    moduleName = "Copy Not Known Files"

    def getName(self):
        return self.moduleName

    def getDescription(self):
        return "Copy Not Known Files,”

    def getRelativeFilePath(self):
        return "hashes.csv"
```

My example plugin

- There must be a log, which will allow us to see the outputs.
- Can check it going to the menu
  - Tool »
    See log file

```
from java.util.logging \
    import Level
from org.sleuthkit.autopsy.coreutils \
    import Logger

class NotKnownBackup(
    GeneralReportModuleAdapter):
...
    _logger = None
    def log(self, level, msg):
        if self._logger == None:
            self._logger = \
                Logger.getLogger(
                    self.moduleName)


        self._logger.logp(
            level,
            self.__class__.__name__,
            inspect.stack()[1][3], msg)
```

My example plugin

- Main process is done in the function **`GenerateReport`**
- Open the file in the directory of the report.
- Using **utf8**, avoids conflicts with unicode file names.

```
class NotKnownBackup(
    GeneralReportModuleAdapter):
...


    def generateReport(self,
                       baseReportDir,
                       progressBar):

        fileName = \
            os.path.join(baseReportDir,
            self.getRelativeFilePath())

        report = codecs.open(fileName,
                 'w', "utf8")
```

My example plugin

- Instantiate the cause **`sleuthkitCase`**
- Creates a list with all the files that are **not** Directory-Type.

```
def generateReport(self,
                   baseReportDir,
                   progressBar):
    ...

    sleuthkitCase = Case.\
        getCurrentCase().\
        getSleuthkitCase()

    files = sleuthkitCase.\
        findAllFilesWhere(
        "NOT meta_type = " +
        str(TskData.
            TSK_FS_META_TYPE_ENUM.
            TSK_FS_META_TYPE_DIR.
            getValue()))
```

My example plugin

- Creates a directory for files whose content copy.
- Creates other subdir for those unknown MIME type.

```
def generateReport(self,
                   baseReportDir,
                   progressBar):
    ...
    if not os.path.exists(
            config.output_path):
        os.mkdir(output_path)

    defaultcontentDir = \
        os.path.join(output_path,
                     "Other")

    if not os.path.exists(
            defaultcontentDir):
        os.mkdir(defaultcontentDir)
```

My example plugin

- In a loop, we go through each file according to its MIME type.
- Defines where it is going to copy it

```
for idx, file in enumerate(files):
    if file.MIMEType:
        typedir = \
            file.MIMEType.\
                replace("/", "_")

        contentDir = \
            os.path.join(
                output_path,
                typedir)
    else:
        typedir = "other"
        contentDir = \
            defaultcontentDir
```

My example plugin

- Write a line with interesting data filein the report file.
- **IsKnown** has true if the file is **well-known**.

```
id = "%12d" % file.getId()

filepath = os.path.join(
            contentDir,
            id + "-" + file.getName())

isKnown = (file.getKnown() ==
           TskData.FileKnown.UNKNOWN)

line = [typedir,
        file.getName(),
        file.getParentPath(),
        str(file.getId()),
        str(file.getMd5Hash())]
```

My example plugin

- Copy the content in the corresponding subdirectory according to MIME type.
- If an error is issued, add it to the **output log**.

```
if not isKnown:
    try:
        if not os.path.exists(
                contentDir):
            os.mkdir(contentDir)

        ContentUtils.writeToFile(
            file, File(filepath))

        report.write(u','.join(line)
                     + "\n")

    except:
        self.log(Level.WARNING,
            str(sys.exc_info()[0]) + "-" +
            str(sys.exc_info()[1]) + "\n" +
            u',.join(line))
```

My example plugin

Finally,

- Closes the file
- Using **`AddReport`** adds the report file to the Autopsy case.

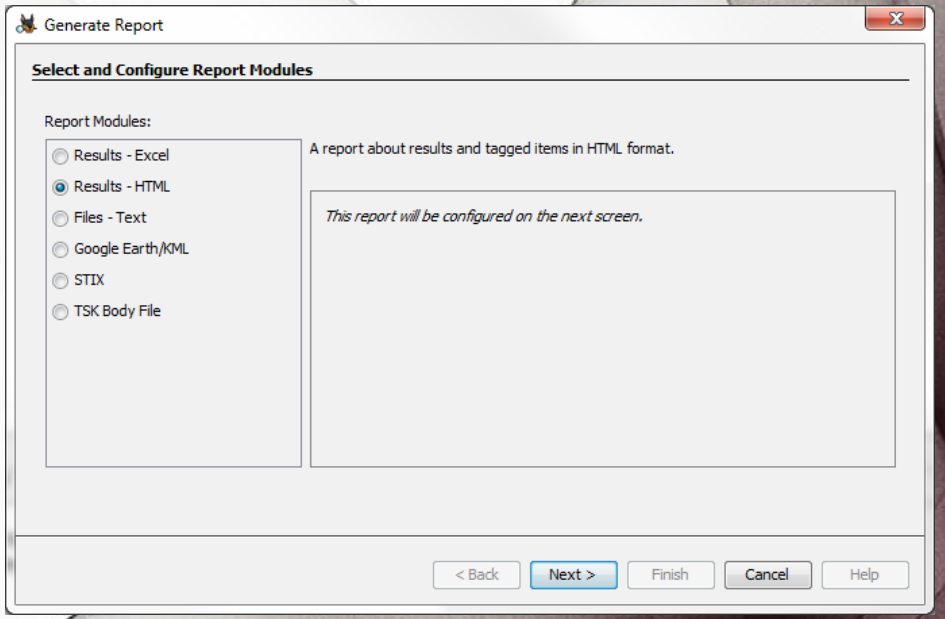
```
class NotKnownBackup(
    GeneralReportModuleAdapter):
    ...


    def generateReport(self,
                       baseReportDir,
                       progressBar):
        ...
        report.close()
        Case.getCurrentCase().\
            addReport(
            fileName,
            self.moduleName,
            "Copy Not Known Files")
```

My example plugin

- Use

  Tools »
  Generate Report

- It is added to the list of available report modules



Computer autopsies

1) Get evidence
2) Make forensic copies
3) Data analysis with Autopsy
4) Extending Autopsy with Python
5) My example plugin.

Computer autopsies

by María Andrea Vignau

Developer and System Engineer
Forensic Criminal Expert
Judicial Branch of the Chaco

Twitter: **@mavignau**
Telegram: **@mavignau**
GitHub: **marian-vignau**