

Autopsias informáticas

Cómo realizar pericias forenses
utilizando herramientas de fuente
abierta y desarrollar plugins de
Autopsy

Ing María Andrea Vignau

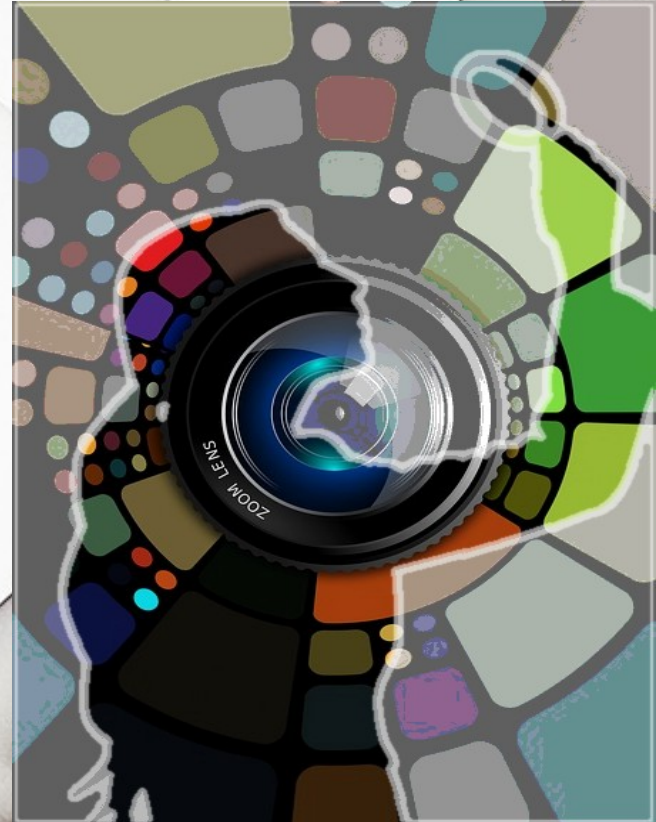
Perito Forense Penal
Poder Judicial del Chaco

Autopsias informáticas

- 1) Obtener evidencia
- 2) Realizar copias forenses
- 3) Análisis de los datos con **Autopsy**
- 4) Extendiendo **Autopsy** con Python
- 5) Mi plugin de ejemplo.

Obtener evidencia

- Identificar dispositivos con capacidad de almacenamiento
- Fotografar
- Ver si están prendidos o apagados
- Evaluar captura de RAM



Obtener evidencia

- Secuestro
 - Es el caso general
- Copia forense parcial en el sitio
 - Servidores especiales



Obtener evidencia

Preservación

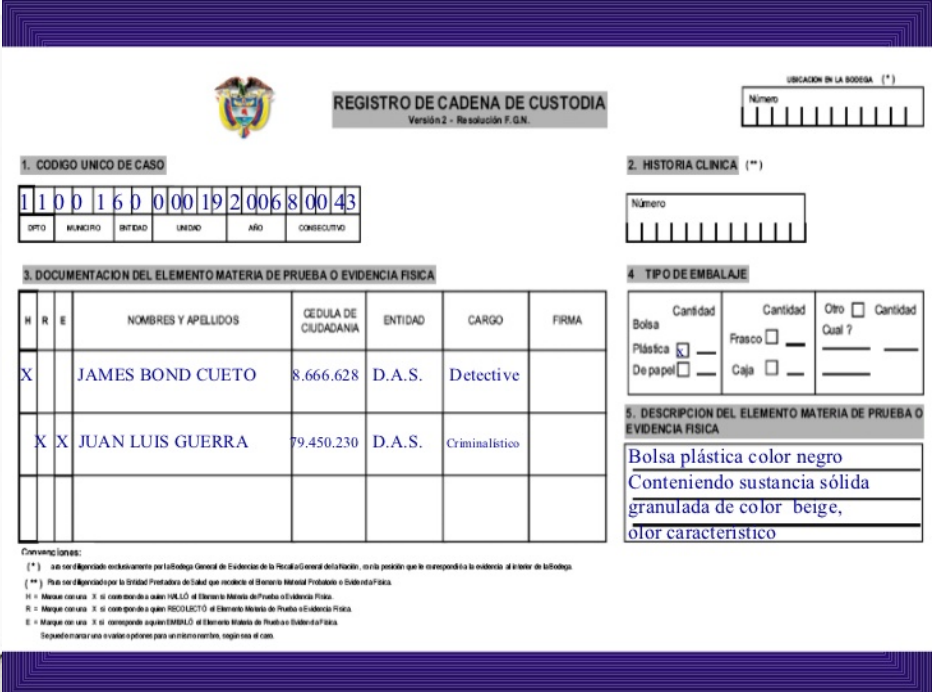
- **Evitar** golpes, humedad
- **Envolver** impidiendo acceso a puertos, o desarmes,
- **Firmar** papel envoltorio.



Obtener evidencia

Cadena de custodia

- Contiene cada persona que se hizo responsable de la integridad de la evidencia.



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

UBICACION EN LA BOBOSA (*)

Número

1. CODIGO UNICO DE CASO

1	1	0	0	1	6	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	GEDULA 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	Cantidad	Otro <input type="checkbox"/> Cantidad
Plástica <input checked="" type="checkbox"/>		Frasco <input type="checkbox"/>	Qual ? <input type="checkbox"/>
De papel <input type="checkbox"/>		Caja <input type="checkbox"/>	

5. DESCRIPCION DEL ELEMENTO MATERIA DE PRUEBA O EVIDENCIA FISICA

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

Convenciones:
(*) No se diligencie exclusivamente por el Subdirector General de Evidencia de la Fiscalía General de la Nación, en la posición que le correspondiera la evidencia al nivel de laboratorios.
(**) Para con diligenciar por la Unidad Prestadora de Salud que emite el Sistema Nacional Probatorio e Subordinado Pánel.
H = Marcar con una X el correspondiente a cada UNIDAD del Sistema Nacional de Pruebas e Evidencia Física.
R = Marcar con una X el correspondiente que RECIBIÓ el Elemento Materia de Prueba o Evidencia Física.
E = Marcar con una X el correspondiente que EMPLAZÓ al Elemento Materia de Prueba o Evidencia Física.
Se pueden marcar una o varias opciones para un mismo nombre, según sea el caso.

Realizar copias forenses



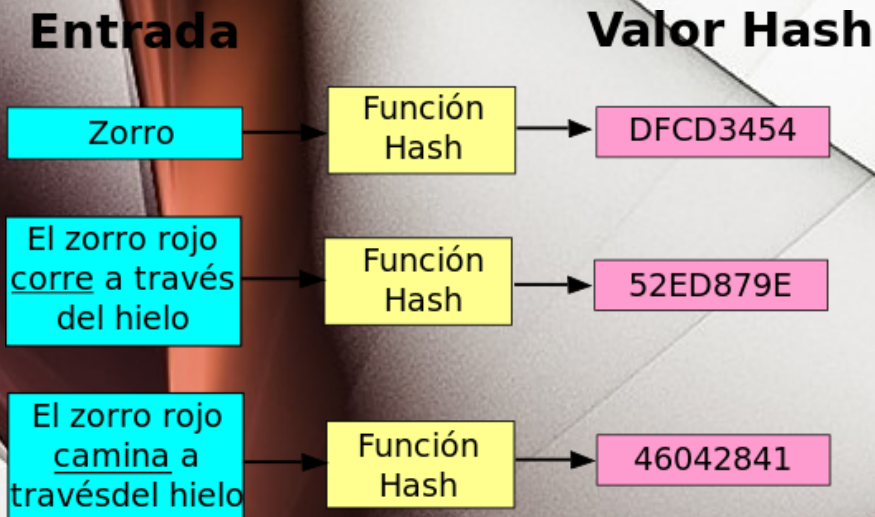
- Copia espejo, bit a bit, incluso espacios no visibles para el usuario.
- Se evita **riesgos** que podrían resultar de procesar directamente la evidencia.

Realizar copias forenses

Asegura la integridad de los datos, de la evidencia.

Función hash:

- **Entrada:** un conjunto de elementos, *cadena*s, y
- **Salida:** un rango finito
- Proyección del conjunto U sobre el conjunto M



Realizar copias forenses



Removiendo el disco rígido.

- Usar duplicador forense
 - Tableau TD3
- Conectar a otra PC, con adaptador a USB
 - Configurar sólo lectura
 - Por Hardware o Software

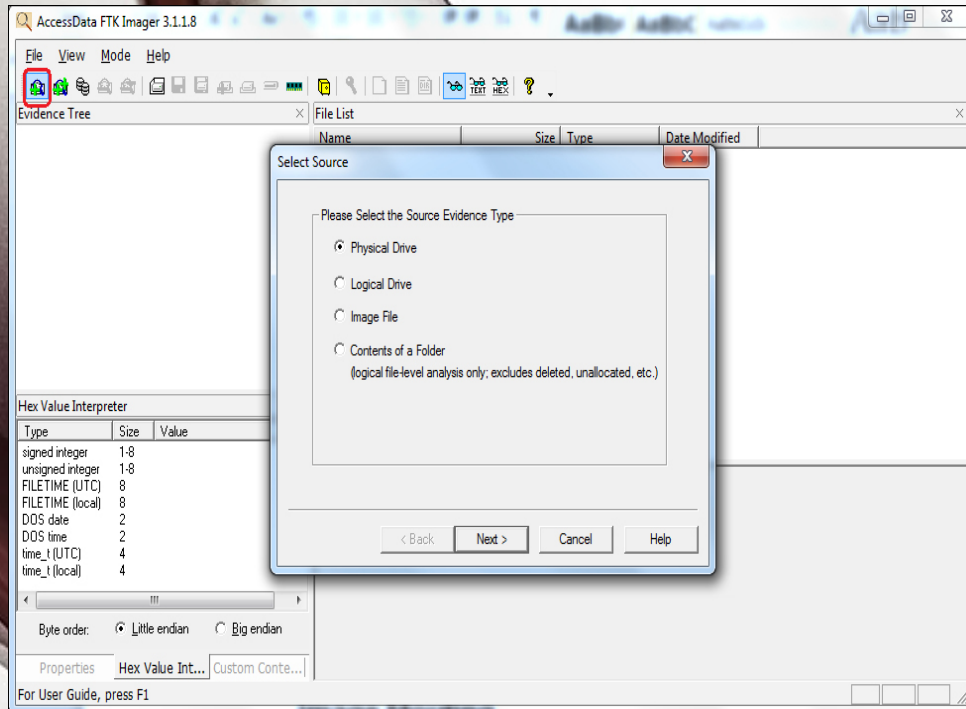
Realizar copias forenses



Windows

- Dejar los puertos USB como sólo lectura
- **USB Disk Access Manager.**

Realizar copias forenses



Windows

- Realizar copia forense
 - **FTK IMAGER**
- Usar formato forense
 - **Expert witness format**

Realizar copias forenses



Sin extraer el disco rígido:

- **Boot** desde disco óptico o pendrive.
- Usar una distribución **especializada**.
- Usar modo forense **sólo lectura**

Realizar copias forenses



Descubrir dispositivos

- `Fdisk -l`

Montar sólo lectura

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

Realizar copias forenses



Linux

- Realizar copia forense
 - **GuyMager**
- Usar formato forense
 - **Expert witness format**

Análisis de evidencia

Autopsy SleuthKit v4.8

- Open Source
- Extensible
- Maduro (v1.0 a 2001)
- Multiplataforma
- Multiusuario



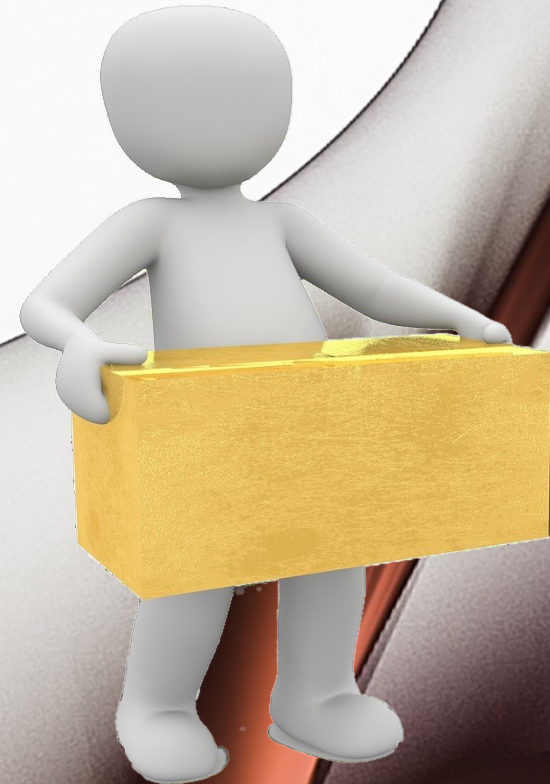
Análisis de evidencia

- Crear un nuevo caso
- Agregar evidencia
- Análisis automático
- Análisis manual
- Reportes



Análisis de evidencia

- **Crear un nuevo caso**
 - Ingresar datos básicos
- **Agregar evidencia**
 - Dispositivo
 - **Imagen forense**
 - Otros



Análisis de evidencia

- Sistemas de archivos
- Imágenes forenses
- Archivos comprimidos
- Carving
- Máquinas virtuales

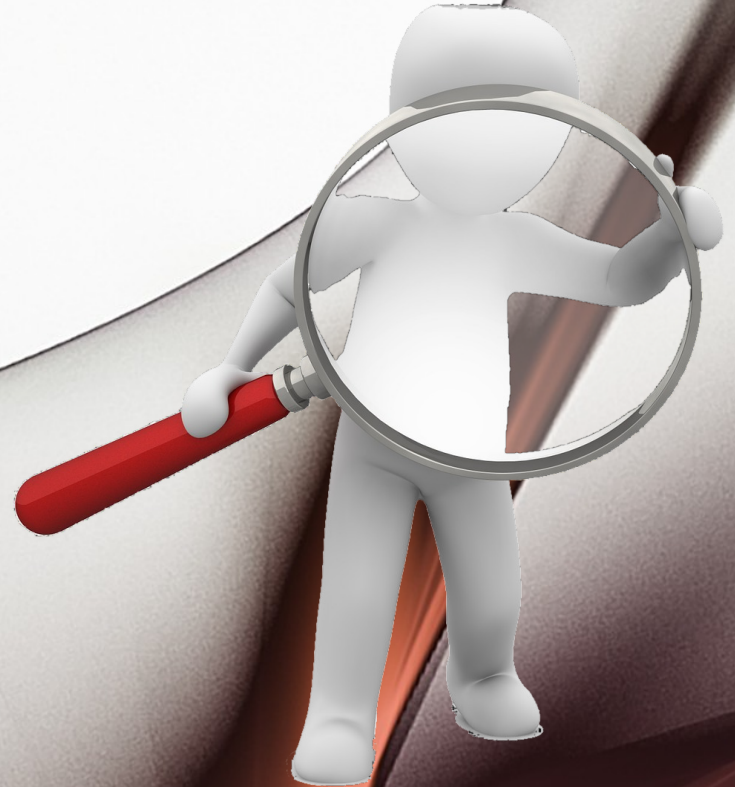
Fuentes de datos procesados



Análisis de evidencia

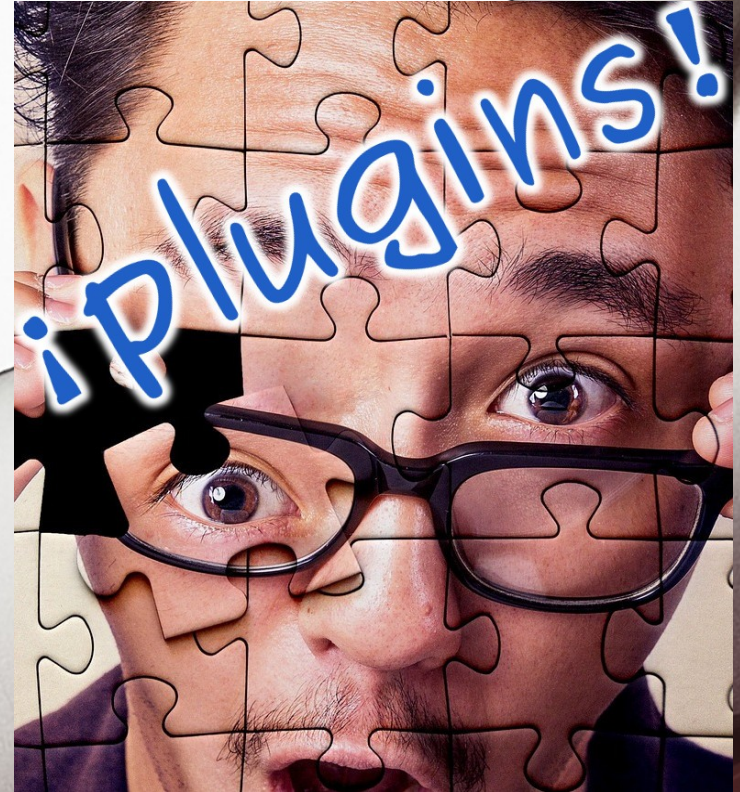
Procesar

- Seleccionar **plugins**
- Cada uno revisa especializadamente la evidencia



Análisis de evidencia

- Hacen hash de la evidencia
- Identifican tipos MIME por su file signature
- Descomprimen archivos
- Sacan imágenes incrustadas en documentos
- Parsean datos de navegadores



Análisis de evidencia

Análisis manual

- Revisar manualmente
- Etiquetar los elementos según el objeto de la investigación

The screenshot displays the Autopsy 3.1.2 interface with several key components highlighted:

- Keyword Search:** A yellow box highlights the search bar at the top right.
- Tree Viewer:** A green box highlights the left-hand navigation pane showing a hierarchical view of data sources and results.
- Result Viewer:** A blue box highlights a table of EXIF metadata for image files. The table includes columns for Source File, Date Created, Device Model, and Device Make.
- Content Viewer:** A red box highlights a preview of a photograph showing a person on a white horse in a city street.
- Status Area:** A purple box highlights the bottom right corner of the interface.

Source File	Date Created	Device Model	Device Make
100_6228.JPG	2011-10-25 06:27:23 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6184.JPG	2011-10-25 05:09:12 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6290.JPG	2011-10-25 10:58:19 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6223.JPG	2011-10-25 06:24:43 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
12-198241.LG.VX8350.5.jpg	2011-09-06 23:35:39 EDT	Canon PowerShot SX110 IS	Canon
12-198241.LG.VX8350.1.jp	2011-09-06 23:35:39 EDT	Canon PowerShot SX110 IS	Canon
100_6594.jpg	2011-10-25 10:58:19 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6418.JPG	2011-10-25 10:03:16 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6342.JPG	2011-10-25 10:58:19 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6290.JPG	2011-10-25 10:58:19 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6259.JPG	2011-10-25 10:03:16 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6228.JPG	2011-10-25 06:27:23 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6223.JPG	2011-10-25 06:24:43 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP
100_6192.JPG	2011-10-25 05:19:00 EDT	KODAK Z650 ZOOM DIGITAL CAMERA	EASTMAN KODAK COMP

Análisis de evidencia

Reportes

- Provistos
- Extensibles
- En formatos diversos.
 - HTML, PDF, etc



Extendiendo Autopsy con Python



- Framework **SleuthKit**
- Desarrollado en **JAVA**
- Extensible usando
 - Java
 - **Python**

Extendiendo Autopsy con Python



Implementaciones de Python

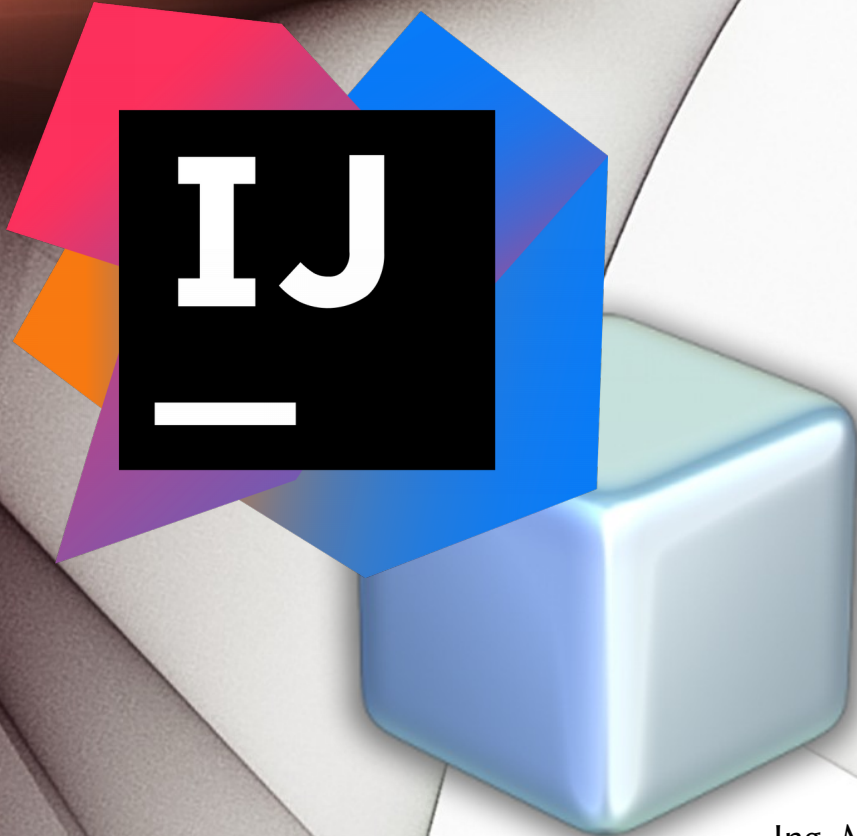
- **Cpython:** Escrita en C.
- **Jython:** en Java
- PyPy
- IronPython ... etc

Extendiendo Autopsy con Python



- Configurar IDE
- Crear esqueleto
- Elegir tipo de módulo
- Elegir forma de salida
- Copiar y adaptar modelo tutorial.

Extendiendo Autopsy con Python



Entorno de desarrollo: IDEs

- IntelliJ IDEA
- NetBeans

Extendiendo Autopsy con Python

Formas de salidas

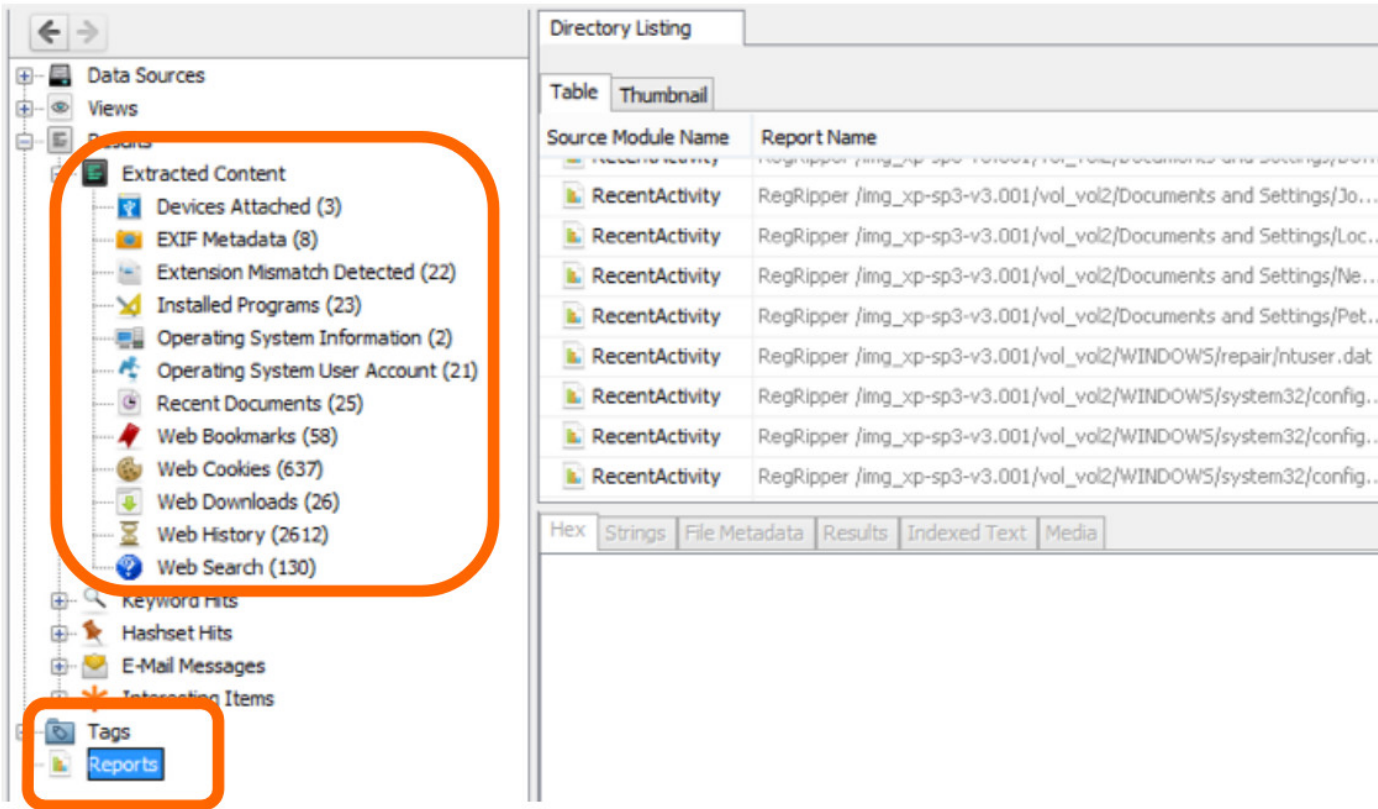


- **Reporte:** el más simple
- **Artefactos** en el Pizarrón.
 - Tipo
 - Archivo asociado
 - Atributos: pares de
 - Nombre, Valor

Extendiendo Autopsy con Python

Artefactos

Reportes



The screenshot displays the Autopsy interface. On the left, the 'Data Sources' pane shows a tree view of artifacts. The 'Extracted Content' folder is highlighted with an orange circle, containing items like 'Devices Attached (3)', 'EXIF Metadata (8)', 'Extension Mismatch Detected (22)', 'Installed Programs (23)', 'Operating System Information (2)', 'Operating System User Account (21)', 'Recent Documents (25)', 'Web Bookmarks (58)', 'Web Cookies (637)', 'Web Downloads (26)', 'Web History (2612)', and 'Web Search (130)'. Below it, the 'Reports' folder is also highlighted with an orange circle. On the right, the 'Directory Listing' pane shows a table of artifacts. The table has two columns: 'Source Module Name' and 'Report Name'. The table contains several rows, all with 'RecentActivity' as the source module name and various file paths as report names. Below the table, there are tabs for 'Hex', 'Strings', 'File Metadata', 'Results', 'Indexed Text', and 'Media'.

Source Module Name	Report Name
RecentActivity	RegRipper /img_xp-sp3-v3.001/vol_vol2/Documents and Settings/Jo...
RecentActivity	RegRipper /img_xp-sp3-v3.001/vol_vol2/Documents and Settings/Loc...
RecentActivity	RegRipper /img_xp-sp3-v3.001/vol_vol2/Documents and Settings/Ne...
RecentActivity	RegRipper /img_xp-sp3-v3.001/vol_vol2/Documents and Settings/Pet...
RecentActivity	RegRipper /img_xp-sp3-v3.001/vol_vol2/WINDOWS/repair/ntuser.dat
RecentActivity	RegRipper /img_xp-sp3-v3.001/vol_vol2/WINDOWS/system32/config...
RecentActivity	RegRipper /img_xp-sp3-v3.001/vol_vol2/WINDOWS/system32/config...
RecentActivity	RegRipper /img_xp-sp3-v3.001/vol_vol2/WINDOWS/system32/config...

Extendiendo Autopsy con Python

Módulo de proceso de ficheros



- Recibe y analiza contenido **cada fichero** en fuentes de datos agregados al caso.

Extendiendo Autopsy con Python

Módulo de proceso de fuente de datos



- Si conocemos dónde estará el fichero
- Con herramienta externas
- Refiere a una fuente de datos **entera**.

Extendiendo Autopsy con Python

Módulos de Reportes



- Corre después del análisis para crear una salida de reporte.
- Se puede usar datos de ficheros, artefactos y etiquetados por usuario
- HTML, XML, CVS

Extendiendo Autopsy con Python

¿Qué tipo de módulo me conviene?



- ¿Debo recorrer cada fichero?
- ¿Sé con precisión qué fichero busco?
- ¿Debo correrlo al final, luego del análisis manual?

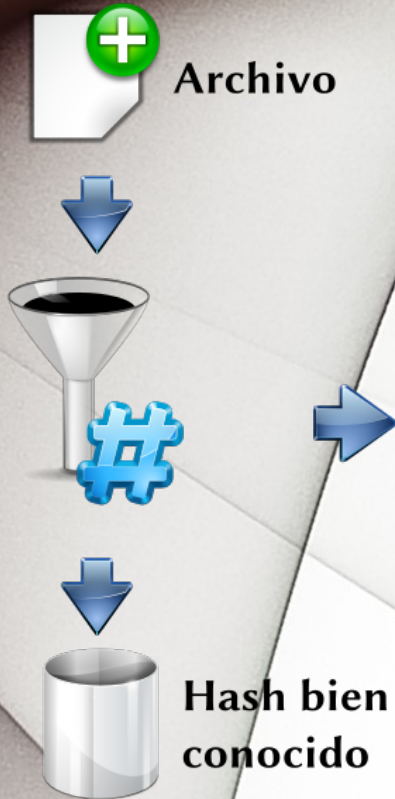
Mi plugin de ejemplo

- Su hash se encuentra identificado.
- Bases de datos en el NSRL del NIST para Autopsy

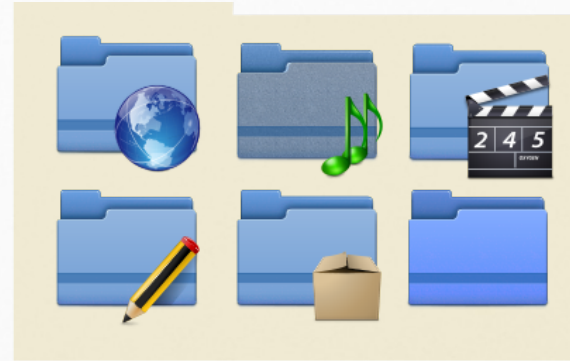
Archivos bien conocidos

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

Mi plugin de ejemplo



Copio
contenido
clasificado
por tipo
MIME



Agrego los datos
al archivo CSV

Mi plugin de ejemplo

- Generamos la clase de nuestro reporte heredando de `GeneralReportModuleAdapter`
- Configuramos las propiedades nombre, descripción y path al archivo de salida

```
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"
```

Mi plugin de ejemplo

- Debemos realizar un log, que nos va a permitir ver las salidas.
- Lo vamos a poder revisar yendo al menú
 - 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)
```

Mi plugin de ejemplo

- El proceso se realiza en la función **GenerateReport**
- Abrimos el archivo en el directorio del reporte.
- Usamos **utf8**, evita conflictos con nombres de archivo unicode.

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

    def generateReport(self,
                       baseReportDir,
                       progressBar):

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

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

Mi plugin de ejemplo

- Instanciamos la causa en **sleuthkitCase**
- Armamos una lista con todos los ficheros que no sean de tipo Directorio.

```
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()))
```

Mi plugin de ejemplo

- Creamos un directorio para los archivos cuyo contenido copiamos
- Creamos un archivo para los que tienen tipo MIME desconocido.

```
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)
```

Mi plugin de ejemplo

- En un bucle, recorreremos cada archivo según su tipo MIME.
- Definimos dónde lo vamos a copiar

```
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
```


Mi plugin de ejemplo

- Armamos una línea con datos que nos interesan en el archivo de reporte.
- **IsKnown** tiene verdadero si el archivo es bien conocido.

```
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())]
```

Mi plugin de ejemplo

- Grabamos el contenido en el subdirectorio que le corresponde según tipo MIME.
- Si hay error, lo agregamos al log de salida.

```
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))
```

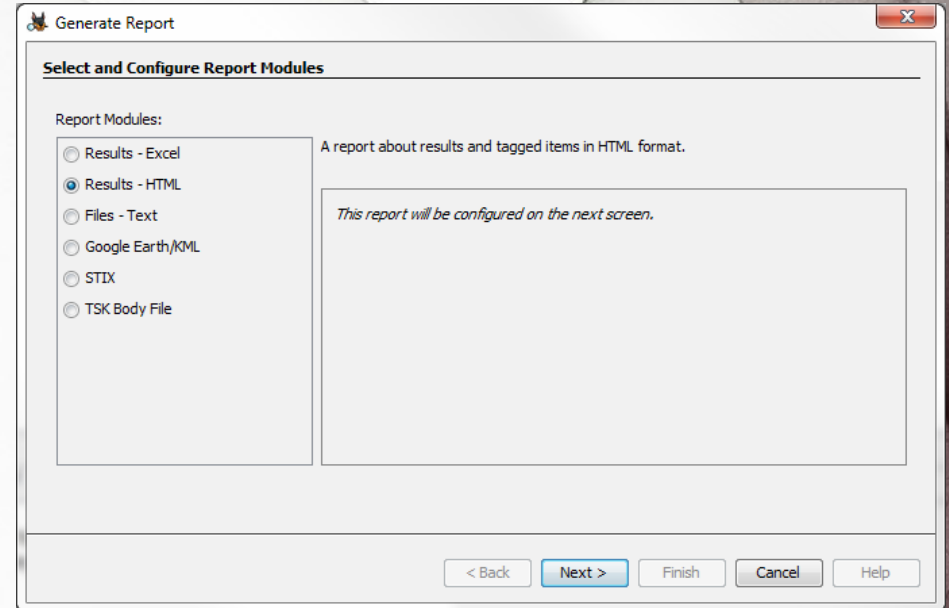
Mi plugin de ejemplo

- Finalmente, cerramos el archivo y usando **AddReport** agregamos el archivo del reporte al caso Autopsy.

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

Mi plugin de ejemplo

- Ir a
Tools »
Generate Report
- Se agrega a la lista de módulos de reporte disponibles



Autopsias informáticas

- 1) Obtener evidencia
- 2) Realizar copias forenses
- 3) Análisis de los datos con **Autopsy**
- 4) Extendiendo **Autopsy** con Python
- 5) Mi plugin de ejemplo.

Autopsias informáticas

por **María Andrea Vignau**

Ing en Sistemas de Información
Perito Informático Forense
Poder Judicial Chaco

Twitter: **@mavignau**

Telegram: **@mavignau**

GitHub: **marian-vignau**

