FEA@Autopsy

João Mota // 2161328 // Projeto Informático

Forensics Enhanced Analysis

# Background

Out of the many products available in the domain of Digital Forensics, the Sleuth Kit (<https://sleuthkit.org>) is one of the best known sets of tools that provide industry standard techniques for undertaking many common tasks, with the Autopsy application providing a unified GUI for it.

As a free-to-use open-source product, Autopsy has had strong support from users and contributors, and has been kept up to date with the latest techniques and best practices of the industry, greatly due to community-driven development, made easy by its modular architecture based in Java and Python, that allows for rapid extensibility.

One of the standard features of the Sleuth Kit exposed by Autopsy is the Keyword Search module, which includes specific features for identifying email addresses and credit card numbers retrieved from data sources - but has a sufficiently generic architecture to allow it be used in several different types of searches.

The initial purpose of this project was to specifically improve the email address search functionality included by default in Autopsy, which is to be achieved by creating additional layers of validation to the standard modules, in order to reduce the currently high number of false positives it yields. However, throughout the development of this feature, it became apparent that other relevant contributions could be made by leveraging the same knowledge garnered in the process, and two additional modules were built: one for finding and validating credit card account numbers, and another for dealing with potential Bitcoin wallet addresses.

It is expected that these tools will provide convenient assistance to forensics experts using Autopsy, as they all target data that is typically of high relevance within this field of work, especially when dealing with suspected fraud, financial crimes, tax evasion and so on.

# Related work

Given Autopsy’s open architecture, there are many community-drive efforts to develop modules for varied purposes, that greatly enrich and extend the default module suite provided by the Sleuth Kit. These are split into three main stages of the forensics analysis process: ingestion, visualization and reporting.

Most of the modules available focus on the ingest phase, which makes sense as it is the most time-consuming and resource-intensive stage, and therefore can bring the most added value through optimization. The concept is to extract only the data that is relevant to the current investigation, and so many different modules are available, trying to cater for many common needs within the field. One of the most flexible ingest modules is included by default in the Sleuth Kit, and is called “Keyword Search”. This module includes a few default keyword lists to facilitate searching the selected data sources for phone numbers, IP addresses, URLs, Credit Card Numbers, and Email addresses. Nevertheless, these searches are basically filters for regular expressions, and incorporate no further validation of the artefacts gathered from the data sources – and even though it’s possible to customize and add keyword lists by building more regular expressions or word lists, the possibilities are rather limited in terms of in-depth validation.

Data visualization modules, referred to as Data Content Viewer Modules, are also quite popular, and provide convenient ways to analyse artefacts gathered from data sources directly. These range from simple text viewers to video triage modules that facilitate viewing large video files by analysing their contents. Again, these are of no help for performing deep validation of email addresses and similar data, as they target individual files and not bulk processing of multiple artefacts.

Lastly, the report modules are seemingly a distant relative of the rest, since there are few contributions from the community, and as far as it was possible to determine, only the standard Sleuth Kit ones are available. Granted that those can satisfy the most common needs of reporting, but the fact is that their capacity to drill down on ingested data is clearly being underused, even though the Sleuth Kit’s documentation tries to raise awareness to the fact that these aren’t meant just for producing information briefs, but can also provide a convenient vehicle for new tools to assist in the investigation process.

After careful consideration of the goals of this project and the framework provided by Autopsy, it was decided that report modules coupled with custom keyword lists in the Keyword Search ingest module would incorporate the most adequate approach. This separation of scope means that the ingest stage doesn’t have to be further burdened with further resource-intensive tasks, allowing the investigator to cherry-pick the most relevant artefacts to be submitted to the report modules developed for the abovementioned specific purposes, with the added extra of allowing for the automatic creation of multiple reports in file formats that facilitate further manual processing.

* Tipos de módulos
* ModuleSettings (config)
* <https://www.experts-exchange.com/questions/28595675/asynchronous-dns-lookup.html>

1. Resumo (PT)
2. Abstract (EN)
3. Índice
4. Acrónimos (ordem alfabética!)
5. Introdução – apresentação do Autopsy, objetivos, motivação, contributos (email, CC, BC)
6. Trabalhos relacionados (ingest modules e plugins já existentes, estrutura modular do autpsy, módulos existentes n standard, concluir que nenhum cumpre os objetivos do atual trabalho)
7. Email Module
8. CC
9. BC
10. Conclusão
    1. Trabalho futuro
11. Bibliografia

# Autopsy Modules

*Autopsy was designed to be an extensible platform for other developers to leverage. There are several places in the platform where plug-in modules can be applied.*

* *Ingest Modules: These modules are run when a new data source (such as a disk image) is added to a case (and can be re-run afterwards too). These modules come in two forms:*
  + *File Ingest Modules are called for every file in the data source. Use this type of module if you want to examine the contents of all or most of the files. Examples include hash calculation, hash lookup, file type identification, and entropy calculation. These modules are passed in a reference to a file to analyze.*
  + *Data Source Ingest Modules are called once for every image or set of logical files. These modules can use the database to query for one or more files and perform analysis on them. Examples include web artifact analysis and searches that can rely only file names and extensions. Note that these modules will not have access to the contents of ZIP files. These modules are also often used when wrapping an executable that takes a disk image in as input. See Developing Ingest Modules for details on building these modules.*
* *Report Modules: These modules are (typically) run after the user has reviewed results and tagged files. Their intention is to create an output report of the results, but they can also be used to perform analysis. See Developing Report Modules for details on creating these modules.*
* *Content Viewers: These modules are graphical and focus on displaying a specific file in some unique way. These are the modules in the lower right of the interface. The platform comes with viewers to view the file in hexadecimal, extract the strings from the file, and view images and movies. See Developing Content Viewer Modules for details on creating these modules.*
* *Result Viewers: These modules show information about a set of files. These modules are in the upper right of the interface. The platform comes with viewers to view the set of files in a table and thumbnails. These are not commonly built and extended. See Developing Result Viewer Modules for details on creating these modules.*

(Basis Technology, 2012-2016)

One of the first decisions to take when addressing the current challenge was which kind of Autopsy module to create, since there are two immediate possibilities: an ingest module or a report module.

Writing an ingest module would have the advantage of increased processing speed, as the script would be able to quickly discard false positives without spending time or resources adding them to the results list. Nevertheless, I’ve decided to go for a report module based on the existing keyword search module, assuming that being a core feature of this software suite means that it is already optimized in the way it tackles the task of doing the actual ingestion, and mostly to take advantage of any future updates that may be introduced by the community to that basic functionality.

* 1. Keyword Search Module

<http://sleuthkit.org/autopsy/docs/user-docs/3.1/keyword_search_page.html>

* 1. The Blackboard

The Sleuth Kit framework’s Blackboard is a collection of “artifacts”, each consisting of name-value pairs called “attributes”, which are used to share information between modules.

Each module can post or get data from the Blackboard, with the purpose of collaboratively solving a problem, and the standard Autopsy modules are no exception.

In order to access the Blackboard from a Jython script, both the *SleuthkitCase* object and the *Content* object can be used. In the former case, a comprehensive list of the Blackboard’s artifacts can be retrieved through one of the many getBlackboardArtifacts methods.

The email address searching feature of the keyword search module in particular puts results into the Blackboard as artifacts containing the attribute TSK\_SET\_NAME with the value “Email Addresses”, we can resort to one such method to retrieve every artifact generated by that feature, and then iterate through each attribute to retrieve the actual email address strings, which are stores in attributes with type TSK\_KEYWORD. A similar approach is used for obtaining the credit card numbers, which differ only in the attribute type added to the result set of artifacts (TSK\_ACCOUNT), and the attribute type of the string value with the actual numbers (TSK\_CARD\_NUMBER),

# The FEA Report Module

* 1. Validating Email Addresses motivação para criação do módulo
     1. TLD Validation

The official TLD database is maintained by IANA ([www.iana.org](http://www.iana.org)), the Internet Assigned Numbers Authority, through its “Root Zone Database” (<https://www.iana.org/domains/root/db>).

The implemented software resorts to scraping the website to retrieve valid TLDs, since there is no public API or any such service available for doing so. This relies on the list made public on <https://data.iana.org/TLD/tlds-alpha-by-domain.txt>. This list is retrieved just before the start of the artifacts processing algorithm, and each time the module is ran, in order to ensure that the latest version is being used.

* + 1. Domain name validation

NSLookup algorithm description (using java.net library)

#TODO: build cache for NS Lookup requests

* + 1. Wayback Machine validation

<https://archive.org/about/terms.php>

* 1. Generated artifacts
* Valid emails
* False positives
* Distinct valid domain names
* Distinct invalid domain names

CC Validation

Bitcoin wallet validation

Hash validation + Simple query API Blockchain

Regular expression to be used in ingestion for public keys (standard keyword search module): “^[13][a-km-zA-HJ-NP-Z1-9]{25,34}$”

For private keys: ^5[HJK][0-9A-Za-z&&[^0OIl]]{49}

<https://en.bitcoin.it/wiki/Private_key>

For checking private keys, the <https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm> is used for translating candidates to public wallet addresses, and validating those instead (using <https://pypi.python.org/pypi/ecdsa#downloads>)

(challenge: converting compressed – beginning with K or L - to uncompressed – which begin with 5)

Used bitaddress.org to generate sample addresses

# References

Apache. (s.d.). *Solr*. Obtido de http://lucene.apache.org/solr/.

Basis Technology. (2012-2016). *Autopsy Forensic Browser Developer's Guide and API Reference*. Obtido de Autopsy 4.1: http://www.sleuthkit.org/autopsy/docs/api-docs/4.3/

Carrier, B. (2011-2013). *The Sleuth Kit (TSK) Framework User's Guide and API Reference*. Obtido de www.sleuthkit.org: https://www.sleuthkit.org/sleuthkit/docs/framework-docs/index.html

Josh Juneau, J. B. (2010). *The Definitive Guide to Jython.* Creative Commons 3.0.

Juneau, J. (03 de 03 de 2011). *Jython 2.5.2 Documentation*. Obtido de http://www.jython.org/docs/index.html

Keffer, J. (2013). Autopsy Forensic Browser User Guide.

Pinto, A. (2017). *Investigação digital com Autopsy.*

Wikipedia. (12 de 03 de 2017). *Luhn algorithm.* Obtido de Wikipedia: https://en.wikipedia.org/wiki/Luhn\_algorithm