sekoia
#CYBERSECURITY

FastIR Collector
Documentation

Sekoia

16/10/2015

SEKOIA Risk Consulting - 16, Place de la Madeleine - 75008 PARIS
Tél. : 01 43 48 25 25 - Fax : 01 47 23 68 14 - E-mail : contact@sekoia.fr

www.sekoia.fr

Content

1. FastIR Collector

FastIR Collector is a “Fast Forensic” acquisition tool. Traditional forensics has reached its limit with the constant evolution of information technology. With the exponentially growing size of hard drives, their copy can take several hours, and the volume of the data may be too large for a fast and efficient analysis.

“Fast Forensic” allows to respond to those issues. It aims a extracting a limited, but with high informational value, amount of data. These targeted data are the most consistent and important ones for an incident response analyst and allows the analyst to quickly collect artifacts and thus, to be able to quickly take decisions about cases.

FastIR Collector is dedicated to the extraction of the most well-known Windows artifact used by different malwares. It helps the analyst to make quick decisions about the status of the acquired system: whether it is compromised or not.

Classic forensic tools need to shutdown systems in order to extract data. FastIR, on the contrary, runs on live systems, without having to turn the system off. This allows investigators to quickly be able to run the tool on systems.

The average execution time of FastIR Collector using the default parameters is about five minutes. Most of the results are outputted under the CSV format.

Currently, FastIR Collecter can analyze the following versions of Windows:

- Windows XP ;
- Windows Vista ;
- Windows 7 ;
- Windows Server 2008 R2.

FastIR Collector is composed of several **analysis packages**, each one being able to retrieve a certain class of artifacts. These packages are presented in detail in the “The profile bloc” part.

FastIR Collector generated data can be analyzed by either the analyst him/herself or a post-processing tool.

2. User guide

FastIR Collector is designed to be easy-to-use. A simple double-click is enough to launch the extraction. The default profile uses the "**fast**" configuration, containing all the fastest packages for the decision making. The "**fast**" configuration includes the following packages:

- registry ;
- memory ;
- evt ;
- fs ;
- health.

Once executed, a folder is created in the folder where the collector is and contains the collector's results.

The "dump" and "filecatcher" packages are not included in the default configuration. It is required to explicitly add them in the configuration in order to execute these packages.

Beware however, since the use of those packages are more time consuming and may take while before finishing.

FastIR Collector is configurable with the use of configuration files called "profiles". It is possible to individually specify which packages should be used and many more options to fine-tune the extraction.

The use of different profiles can be specified using the command line options:

FastIR.exe --profile <profile.conf>

2.1. Creating a profile

A profile file is composed of many blocks of options (between brackets) and individual options.

```
[profiles]
packages=fast
```

*Here, **profiles** is a block, **packages** is an option and its value is "fast"*

The different blocks of options are: profiles, dump, output, filecatcher, modules, pe, yara.

2.1.1. The « profile » block

The profile block is composed of a single option: **packages**.

The packages option

The packages options allows to specify which packages should be executed by FastIR Collector.

The different available packages are:

- registry: the registry extracting package ;
- memory: the in-memory processes extracting package ;
- evt: the log files extracting package ;
- fs: the filesystem extracting package ;
- health: the WMI based extracting package, several different kinds of information are extracted here ;
- dump: the classic forensic artifacts extracting package ;

- filecatcher: the file extracting package, based on several given rules.

The special package "**fast**" regroup the registry, memory, evt, fs and health packages.

It is also possible to manually specify which package should be executed, by separating each package with commas.

Example: `packages=registry,evt,health`

If the "dump" or "filecatcher" packages are specified, it is required to define additional options blocks, respectively "The dump block" and "The filecatcher block".

2.1.2. The « dump » block

The dump block is composed of 2 options, dump and mft_export.

The dump option

It is the most important option in the block. The analyst is able to specify which type of dump should be produced. The different kinds of dump are:

- mft: Master File Table of the NTFS filesystem ;
- mbr: Master Boot Record ;
- ram: RAM of the system ;
- dd: Whole disks.

Beware, the dump of disks should be done on a larger external support and can be time consuming.

It is possible to specify multiple options by separating them with commas.

Example: `dump=mft,mbr,ram`

The mft_export option

This option is only used when the MFT dump option is activated. It can specify the collector's behavior, either by dumping the whole raw MFT file, or by computing the MFT contents and outputting them in a CSV file.

The possible values are **True**, to dump only the raw MFT file, or **False**, to compute the contents of the MFT and generate a CSV output.

2.1.3. The « output » block

The "output" block is composed of 6 options: type, destination, dir, share, share_login and share_password.

The type option

Specifies which type of output should be generated by the collector. As of today, the only available output is **csv**.

The destination option

Unused at the moment.

The dir option

Specifies in which folder the results should be stored. By default, the folder is created in the same path as the one where FastIR Collector is executed. It is possible to specify absolute path (e.g. "C:\output_folder") to modify this behavior. The **share** option also modifies this behavior, by making FastIR Collector create the folder on the specified remote share.

The share option

Specifies a remote network share to store the results. The folder specified in the **dir** option will be created on the network share.

The share_login and share_password options

This option is only used if the **share** option is specified. FastIR Collector can use the specified information to connect to the remote network share if it is protected by credentials.

Since those information are stored in clear text in the file, it is of utmost importance to appropriately protect this profile.

2.1.4. The « filecatcher » block

The filecatcher block is composed of 10 options: path, recursively, mime_filter, mime_zip, compare, size_min, size_max, ext_file, zip_ext_file and zip.

The path option

Specifies where FastIR Collector will execute the filecatcher. It is possible to specify environment variables such as %USERPROFILE, %PUBLIC%, etc.

In order to specify multiple folders to search in, values should be comma-separated.

Example: `path=%USERPROFILE%,%PUBLIC%,C:\tmp`

The recursively option

This option is only used if environment variables that are related to the OS users are specified in the **path** option. Currently supported environment variables are: TEMP, USERPROFILE, APPDATA, LOCALAPPDATA, TMP.

Possible values are either **True**, so that the filecatcher searches through all users, or **False**, where the filecatcher will only search inside the folders of the user currently running FastIR Collector.

The mime_filter and mime_zip options

Defines MIME filters used by the filecatcher.

The **mime_filter** option specifies MIME filters. The absolute path of files corresponding to the specified MIME filters are logged in the output text file.

The **mime_zip** option is similar, files that match the specified MIME filters are instead copied inside an output ZIP archive.

For both options, values should be comma-separated “magic” rules. The wildcard character “*” is supported and tells FastIR Collector to capture every files.

The ext_file and zip_ext_file options

Specifies the files extensions the filecatcher should match.

The **ext_file** option specifies which file extensions should be retrieved. The filecatcher logs the absolute path of all files with matching extensions in the output text file.

The **zip_ext_file** is similar, files with matching extensions are instead copied inside an output ZIP archive.

For both options, values are comma-separated files extensions. The wildcard character “*” is also supported and tells FastIR Collector to capture all files with matching extensions. It is also possible to specify the string “|EMPTY|” in order to collect files with no extension.

The compare option

Specifies which of AND or OR logical operand should be used between the previously described options.

The compared options are the **mime_filter/ext_file** and **mime_zip/zip_ext_file** options.

When the **compare** option is set to AND, FastIR Collector will only collect files satisfying both conditions.

When the **compare** option is set to OR, FastIR Collector will only collect files satisfying either of the 2 options.

The size_min and size_max options

Specifies the minimum and maximum size of the files to be retrieved. Sizes should end with one of the 3 following letters: k (kilo), M (mega) or G (giga).

The zip option

Specifies if FastIR Collector should generate or not a ZIP archive containing all files matching the **mime_zip** and **zip_ext_file** options.

Possible values are **True**, to generate a ZIP archive, or **False**, to not generate a ZIP archive.

2.1.5. The « modules » block

Specifies additional option blocks that will be loaded by the filecatcher. Users should not modify this section.

2.1.6. The « pe » block

The **pe** block is composed of 4 options: pe_mime_type, filtered_certificates, cert_filtered_issuer and cert_filtered_subject.

This block is an additional filter for the filecatcher package.

The pe_mime_type option

Specifies the MIME types on which the certificate should be checked. Default parameters match Windows PE files and should not be changed unless the analyst really knows what he is doing.

The filtered_certificates option

Specifies if certificated based filtering should be applied.

Possible values are **True**, to enable certificate filtering, or **False**, to disable certificate filtering.

The cert_filtered_issuer option

Filters files (mainly executables) depending the certification authority.

Values are certification authorities names separated by pipes (“|”).

The cert_filtered_subject option

Filters executables depending on the certificate subject.

Values are certification subjects separated by pipes (“|”).

2.1.7. The « yara » block

The yara block is composed of 2 options, filtered_yara and dir_rules.

This block is an additional filter for the filecatcher package.

The filtered_yara option

Enables or disables yara rules filtering.

Possible values are **True**, to enable yara filtering, or **False**, to disable it.

The dir_rules option

Specifies the folder containing all the yara rules. It is possible to specify absolute file paths. All “.yar” files located in the folder are taken into account. It is not recursive and does not take into account other yara files in sub-directories, so all yara rules files should be located on the first level of the specified folder. Only one folder should be specified.

3. Limitations

3.1. Operating systems

FastIR Collector does not support the latest versions of Windows, Windows 10 and Windows 2012 Server yet.

Currently, it does not support Linux and Mac OS.