]HackingTeam[

Remote Control System

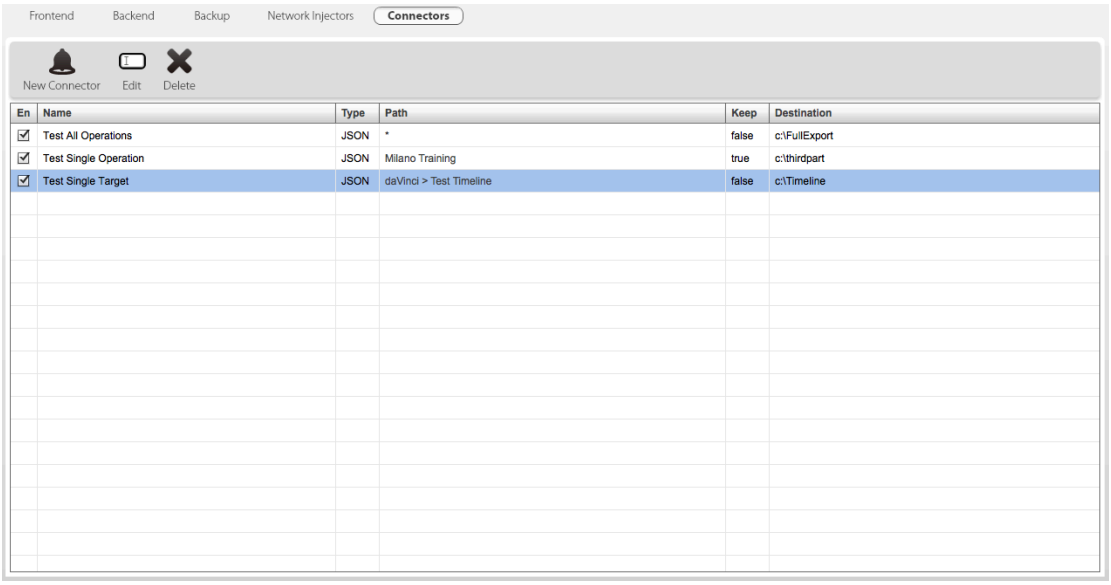
Da Vinci

Integration with third party evidence analyzers

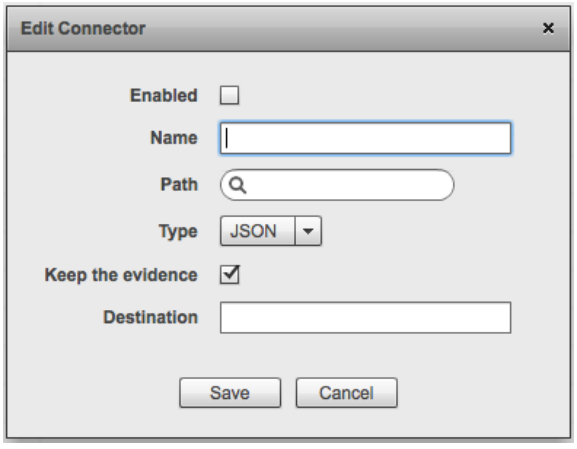
Ver 8.2.0

Configuration:

To configure the RCS system to export the evidence to a third party solution, you must configure a connector. Under the “System > Connectors” section you will find all the rules configured in the system.



It is possible to have different rules for different operations or targets. When you create a new rule a form like this will appear:



Each rule is composed by:

Name: Customizable name for the rule

Path: The operation or target that has to be exported, you can leave it blank to export all the operations

Type: The export type. Currently supported only JSON

Keep the evidence: Controls whether the evidence is stored in the RCS DB or not. If you want to save disk space and you will not use the RCS console to analyze the evidence, you can safely uncheck it.

Destination: The destination directory of the export

Directory structure:

Each exported evidence is stored in a hierarchical structure on the filesystem. The first level is represented by the operation, the second level is the target, the third level is the agent.

Every directory is made up of the name of the object plus the global unique id. Example:

```
operation-4f86902a2afb6512a7000033

      -> target-4f86902a2afb6512a700006f

            -> agent-4fd1a76d2afb65a3cc000039
```

The evidence are saved inside the agent directory and the name of the file is the unique id of the evidence (the same as the _id inside the json format).

For each evidence the connector create one or two files with the same name. The file with extension ".json" is always created and represents the metadata of the evidence. If the evidence has a binary content (es: screenshot, camera, call, etc) another file with the same name is created but the extension will be ".bin". If the "_bin_size" field in the "data" hash of the json file is present, it is a confirmation that the corresponding bin file has been created as well. The bin file contains the raw binary data of the evidence.

JSON Common fields:

Every exported evidence has a common json structure. The common structure is made up of 9 fields:

NAME	**TYPE**	**DESCRIPTION**
_id	Global unique id	A unique id that identifies the evidence itself
da	Unix timestamp	Date of acquisition of the evidence (on the target device) UTC
dr	Unix timestamp	Date of reception of the evidence (on the collector) UTC
aid	Global unique id	Unique id of the agent which generated the evidence
tid	Global unique id	Unique id of the target which generated the evidence
oid	Global unique id	Unique id of the operation containing the target
agent	string	Name of the agent
target	string	Name of the target
operation	string	Name of the operation
type	string	The type of the evidence
rel	integer	The relevance of the evidence (0 to 4)
blo	boolean	Always false in exported evidence
note	string	Always empty ("") in exported evidence
data	hash	The metadata of the evidence, differs based on "type".

Example:


```
{
  "_id": { "$oid" : "4FD9AFD02AFB6514F7000002" },
  "da": 1339658104,
  "dr": 1339658250,
  "aid": "4fd1a76d2afb65a3cc000039",
  "type": "file",
  "rel": 0,
  "blo": false,
  "data": {  },
  "note": ""
}
```


JSON Specific fields:

For each evidence the field “data” may have different fields based on the “type” of the evidence. There may be other fields that are not relevant to the export itself but they can be present. The input parser of the third party solution that import the evidence should be resistant to the addition or deletion of the fields in the exported structure.

ADDRESSBOOK:

FIELD	**TYPE**	**DESCRIPTION**
name	string	The name of the contact
contact	string	The email address
info	string	Extended info about the contact
program	string	The program where the contact was taken from
type	string	The kind of the contact

APPLICATION:

FIELD	**TYPE**	**DESCRIPTION**
program	string	The name of the application
action	string	‘start’ or ‘stop’
desc	string	Description of the application

CALENDAR:

FIELD	TYPE	DESCRIPTION
event	string	The name of the event
begin	unix timestamp	Beginning date of the event
end	unix timestamp	Ending date of the event
Info	string	extended info about the event

CALL:

FIELD	TYPE	DESCRIPTION
peer	string	The peer of the call
program	string	The program from which the call was recorded
status	string	The status of the call (should always be empty)
duration	integer	Duration of the conversation (in seconds)

CAMERA:

FIELD	TYPE	DESCRIPTION
-	-	-

CHAT:

FIELD	TYPE	DESCRIPTION
program	string	Name of the program used for the chat
topic	string	Topic of the conversation
peer	string	The peer(s) of the chat
content	string	The content of the chat

CLIPBOARD:

FIELD	TYPE	DESCRIPTION
program	string	The program that created the clipboard snippet
window	string	Title of the window of the program

content	string	The content of the clipboard

COMMAND:

FIELD	TYPE	DESCRIPTION
command	string	The executed command
content	string	The output of the executed command

DEVICE:

FIELD	TYPE	DESCRIPTION
content	string	Information about the target device

FILE:

FIELD	TYPE	DESCRIPTION
type	string	‘open’ or ‘captured’
program	string	The name of the program that opened the file
path	string	The full path of the file
size	integer	Size of the file (in bytes)
access	integer	Access mask

FILESYSTEM:

FIELD	TYPE	DESCRIPTION
path	string	Path of the filesystem entry
size	integer	Size of the file
attr	integer	0 = file, 1 = empty directory, 3 = directory

INFO:

FIELD	TYPE	DESCRIPTION
content	string	The content of the info log

KEYLOG:

FIELD	TYPE	DESCRIPTION
program	string	The name of the program from which the keys are recorded
window	string	The title of the window
content	string	The recorded keystrokes

MESSAGE:

FIELD	TYPE	DESCRIPTION
type	string	'mail', 'sms' or 'mms'
program	string	The program used to send or receive the message
from	string	The sender of the message
rcpt	string	The receiver of the message
cc	string	Carbon Copy
subject	string	Subject of the message
body	string	Body of the message
attach	integer	Number of attachments

MIC:

FIELD	TYPE	DESCRIPTION
duration	integer	Duration of the recording (in seconds)

MOUSE:

FIELD	TYPE	DESCRIPTION
program	string	Name of the program
window	string	Title of the window
x	integer	Absolute x coordinate
y	integer	Absolute y coordinate
resolution	string	Screen resolution

PASSWORD:

FIELD	TYPE	DESCRIPTION

program	string	The program used to store the credentials
service	string	The service name
user	string	Username
pass	string	Password

POSITION:

FIELD	TYPE	DESCRIPTION
type	string	Type of the source: 'WIFI, 'GPS, 'GSM, 'CDMA'
latitude	float	Latitude
longitude	float	Longitude
wifi	array	Array of hash, each composed by: mac (string): The mac address of the wifi node sig (integer): Current signal strength measured in dBm bssid (string): SSID
cell	hash	Hash composed of: mcc (integer): Mobile Country Code (MCC for GSM and CDMA) mnc / sid (integer): Mobile Network Code (MNC for GSM, SID for CDMA) lac / nid (integer): Location Area Code (LAC for GSM, NID for CDMA) cid / bid (integer): Unique identifier of the cell. (CID for GSM, BID for CDMA) db (integer): Radio signal strength measured in dBm. adv (integer): Represents the distance from the cell tower. age (integer): The number of milliseconds since this cell was primary.

PRINT:

FIELD	TYPE	DESCRIPTION
spool	string	Spool name (typically the file name)

SCREENSHOT:

FIELD	TYPE	DESCRIPTION

program	string	Foreground program name
window	string	Foreground window title

URL:

FIELD	**TYPE**	**DESCRIPTION**
program	string	The name of the browser
url	string	The visited URL
title	string	Window title of the URL page
keywords	string	List of keywords searched on search engines

BIN formats:

The format of the ".bin" file depends on the type of the evidence:

screenshot	Jpeg image
mouse	Jpeg image
camera	Jpeg image
print	Jpeg image
call	Mp3 audio
mic	Mp3 audio
mail	Eml message
file	The original format of the file (can be extracted from the 'path' field in the json metadata)