

]HackingTeam[

REMOTE CONTROL SYSTEM
GALILEO

Integration with third party evidence
analyzers

Ver 9.6.0

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.

En	Name	Path	Type	Format	Keep	Destination	Stat...
<input type="checkbox"/>	Archive Local test	galileo	LOCAL	JSON	true	galileo	0
<input type="checkbox"/>	Archive Local XML	galileo	LOCAL	XML	true	galileo-xml	0
<input checked="" type="checkbox"/>	Archive Test	galileo	REMOTE	RCS	true	172.20.20.152	0

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

NEW CONNECTOR

Enabled

Name

Path

Type LOCAL

Format JSON

Keep the evidence

Destination

Save Cancel

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. Local (saved on filesystem) or Remote (another RCS system).

Format: Currently supported only JSON and XML

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
<code>_id</code>	Global unique id	A unique id that identifies the evidence itself
<code>da</code>	Unix timestamp	Date of acquisition of the evidence (on the target device) UTC
<code>dr</code>	Unix timestamp	Date of reception of the evidence (on the collector) UTC
<code>aid</code>	Global unique id	Unique id of the agent which generated the evidence
<code>tid</code>	Global unique id	Unique id of the target which generated the evidence
<code>oid</code>	Global unique id	Unique id of the operation containing the target
<code>agent</code>	string	Name of the agent
<code>target</code>	string	Name of the target
<code>operation</code>	string	Name of the operation
<code>type</code>	string	The type of the evidence
<code>rel</code>	integer	The relevance of the evidence (0 to 4)
<code>blo</code>	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
handle	string	The unique identifier for that digital identity

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 LIST:

FIELD	TYPE	DESCRIPTION
from	string	The caller
from_display	string	The name of the caller
rcpt	string	The callee
rcpt_display	string	The name of the callee
program	string	The program from which the call was recorded
incoming	integer	Flag to know if the call is incoming or not (0 or 1)
duration	integer	Duration of the conversation (in seconds)

CALL:

FIELD	TYPE	DESCRIPTION
peer	string	The peer of the call
caller	string	Who instantiated the call
incoming	integer	Flag to know if the call is incoming or not (0 or 1)
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
incoming	integer	Flag to know if the chat message is incoming or not (0 or 1)
from	string	The caller
from_display	string	The name of the caller
rcpt	string	The callee
rcpt_display	string	The name of the callee
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
incoming	integer	Flag to know if the message is incoming or not (0 or 1)
attach	integer	Number of attachments

MIC:

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

MONEY:

FIELD	TYPE	DESCRIPTION
type	string	“tx” for cyptocurrency transactions “wallet” for wallet capture

Transactions:

FIELD	TYPE	DESCRIPTION
id	string	The ID of the transaction (the same id as in the blockchain)
from	string	The address of the sender
rcpt	string	The address of the receiver
currency	string	The type of currency used (e.g.: litecoin, bitcoin)
amount	integer	The amount of the transaction
incoming	integer	1 if incoming, 0 if outgoing

Wallet:

FIELD	TYPE	DESCRIPTION
currency	string	The type of currency used (e.g.: litecoin, bitcoin)
program	string	The name of the program that was used to create the wallet
path	string	The path of the wallet file
size	integer	The size of the wallet file

version	integer	The version of the wallet
balance	integer	The amount of coins available in the wallet
encrypted	boolean	Indicates if the wallet is password protected

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

PHOTO:

FIELD	TYPE	DESCRIPTION
program	string	The program used to store the photos library
path	string	The complete path (if present) of the photo
desc	string	Description of the photo
device	string	The device used to take the photo.
tags	array	List of people tagged in the photo. E.g.: [{name: 'pietro gobetti', handle: '1234567890', type: 'facebook'}, ...]
latitude	float	Latitude of the photo
longitude	float	Longitude of the photo
accuracy	int	Radius in meter of the accuracy of the gps coordinates

type	symbol	"target" if the photo contains the target (owner of the library)
------	--------	--

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: <table border="1" data-bbox="571 689 1337 878"> <tr> <td>mac</td> <td>string</td> <td>The mac address of the wifi node</td> </tr> <tr> <td>sig</td> <td>integer</td> <td>Current signal strength measured in dBm</td> </tr> <tr> <td>bssid</td> <td>string</td> <td>SSID</td> </tr> </table>	mac	string	The mac address of the wifi node	sig	integer	Current signal strength measured in dBm	bssid	string	SSID												
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: <table border="1" data-bbox="571 958 1337 1411"> <tr> <td>mcc</td> <td>integer</td> <td>Mobile Country Code (MCC for GSM and CDMA)</td> </tr> <tr> <td>mnc / sid</td> <td>integer</td> <td>Mobile Network Code (MNC for GSM, SID for CDMA)</td> </tr> <tr> <td>lac / nid</td> <td>integer</td> <td>Location Area Code (LAC for GSM, NID for CDMA)</td> </tr> <tr> <td>cid / bid</td> <td>integer</td> <td>Unique identifier of the cell. (CID for GSM, BID for CDMA)</td> </tr> <tr> <td>db</td> <td>integer</td> <td>Radio signal strength measured in dBm.</td> </tr> <tr> <td>Adv</td> <td>integer</td> <td>Represents the distance from the cell tower.</td> </tr> <tr> <td>Age</td> <td>integer</td> <td>The number of milliseconds since this cell was primary.</td> </tr> </table>	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.
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

SYNC:

FIELD	TYPE	DESCRIPTION
ip	string	Ip address of the target
begin	unix timestamp	Date of the beginning of the sync from the agent
end	unix timestamp	Date of the end of the sync from the agent
total	integer	Total number of evidence to be transferred from the agent
count	integer	Actual number of evidence transferred in this sync
speed	integer	Bytes per second
size	integer	The size of the evidence transferred
timeout	boolean	True if the sync has been closed for timeout

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)
money	The original wallet file