DIATEAM

a Cy4gate company

INTEGRITY CHECKER USAGE

EN - 1.1

Reference: DIATEAM-integrity-checker_usage-EN-1.1

DOCUMENT INFORMATIONS

Reference	DIATEAM-integrity-checker_usage
Version	1.1
Nombre de pages	15

DIFFUSION

Recipient(s)	CLIENT

CHANGELOG

Version	Autor(s)	Date	Description
1.0	FBE		File creation
1.1	FBE	28/05/2026	IntegrityChecker config path modification

DIATEAM - 31 rue Yves COLLET - 29200 Brest - FRANCE | Tel : +33(0) 298 050 050

CONTENTS

DIATEAM - 31 rue Yves COLLET - 29200 Brest - FRANCE | Tel : +33(0) 298 050 050

1. PRE-REQUISITE

See [DIATEAM-integritychecker-script] for concept, configuration and scripts usage.

2. USAGE

2.1.1 Check your configuration to catalog

This file contains elements to analyze the catalog filesystem content of your Cyber Range. As it use Hapi scripts, you must configure the service and fill the **config.json** file with your Cyber Range information to permit to check the catalog content.

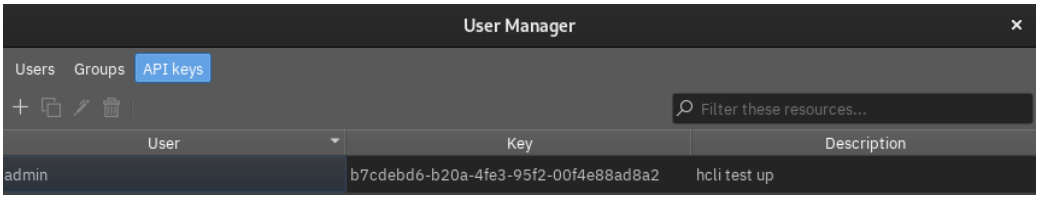
The first version use config.json file located **/usr/share/hynesim/scripts/IntegrityChecker/** folder.

For future version, this configuration file will be relocated to **/etc/integrityChecker/**.

config.json content file :

```
{
        “hapiKey": "YourApiKey”
        “endpoint": "http://api.xxx.diateam.range",
}
```

Change the **X-API-Key** value by the API key of your cyber range. You will find it by hyneview from the **User manager panel provided in the Window** menu :



The endpoint is stored into **/etc/hosts** file. Open the hosts file on Master server, and go to Diateam services section, to know the api link name.

This script can detect :

- **unused entities**
- **orphan entities**
- **entities template present into topologies definition**
- **duplicated uuids into topologies definition**

2.2 SCRIPT USAGE

Project is located into **/usr/share/hynesim/scripts/IntegrityChecker** folder, you can copy the content of **IntegrityChecker** folder on CyberRange.

Open a terminal directly into IntegrityChecker folder.

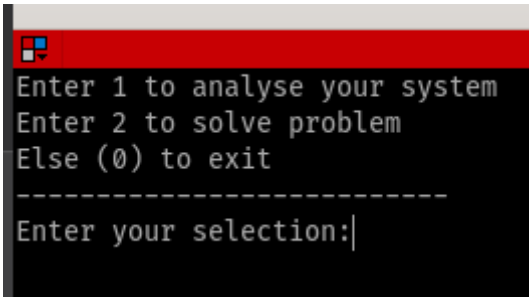
2.2.1 Launch script

Now, you can launch the program.

python3 ./integrityCheck.py

The first menu is displayed and ask you to:

- Analyse your CyberRange
- Resolve problems

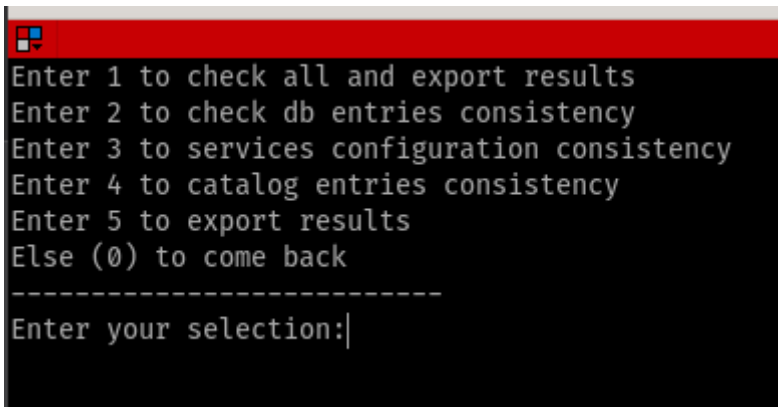


The first thing to do on platform, is to launch a complete analyse of system.

Select **1** to enter into analyse sub menu.

2.2.2 Analyse data

You will see the following menu, that permit us to analyse datas from database, services configuration files, and catalog.



You must launch a full analyse, by selecting **1** in this menu.

Refer to **DIATEAM-integritychecker_script-EN-1.0.pdf** file *$2.2.2 Analyse data*, for explanations about each type of analyse.

For each analyse, 2 cases can be encountered :

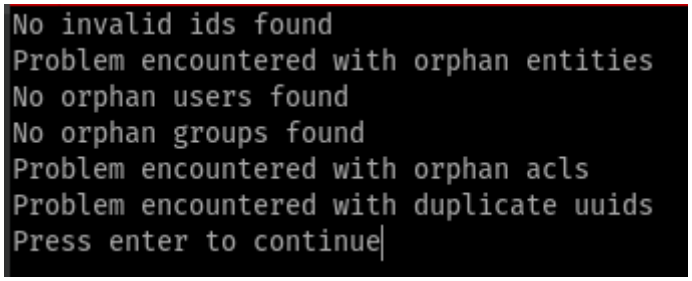
- When no errors were encountered, you can read “No **ResearchInXXX** found”.
- Otherwise, you can read “Problem encountered with **ResearchInXXX**.

The first step is the db analyse.

You will see the result on screen, about problem encountered.

2.2.2.1 Database analyse

The first analyse result concerns database.

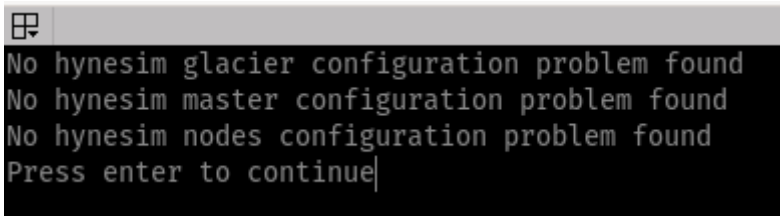


In this example, the only error encountered is orphan acls.

Press enter to view the following result.

2.2.2.2 Service configuration analyse

The second analyse result concerns services configuration.



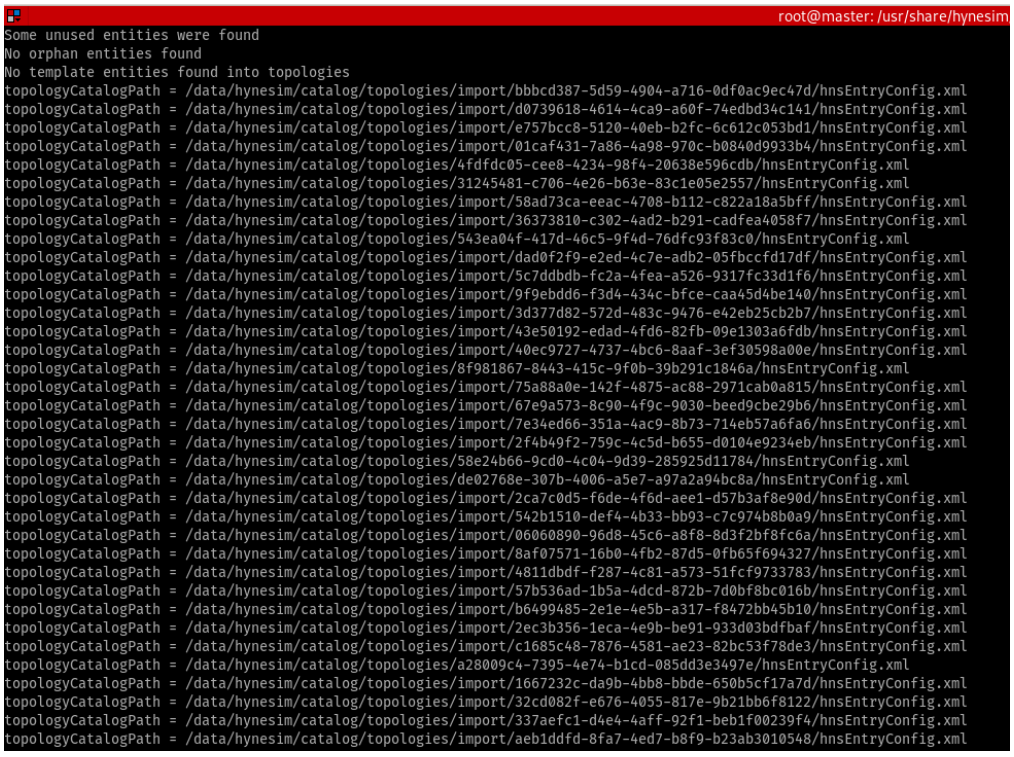
In this example, all hynesim configuration file contains the requested key/value to work correctly.

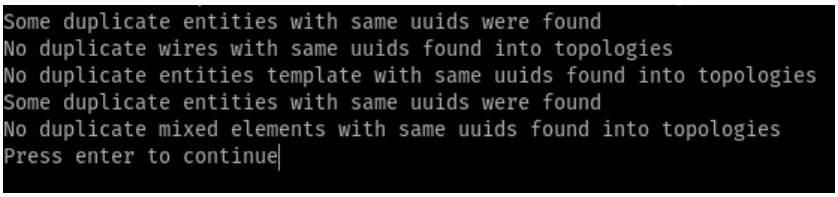
Even if default values are applied, if we want to have a more flexible usage, it is necessary to have these keys, into configuration files.

Press enter to view the following result.

2.2.2.3 Catalog analyse

The last analyze result concerns catalog.





In this example, we have detected :

- some unused entities: It is necessary for platform administrator who want to move/delete some entities. These entities are not used into topologies, and could be removed, to save space on file storage.
- Some duplicated entities into multiple topologies: 2 type of display information. This kind of display is useful, for resolve part:
    - The first one filtered by topologies (the key is the entity uuid, values are topologies uuids where the entities is shared.
    - The second were summaries by topologies uuid all entities uuids encountered.

Press enter to view the analyse result name, and location.

Press enter to continue.

Each time you launch an analyse, if problem were encountered, an analyse report is generated into **/usr/share/hynesim/scripts/IntegrityChecker/analysisResult/** folder.

This folder contains three files :

- DBCheck.xml: is filled by analyse part, and contains only problems encountered by database analyse.
- ServicesConfigurationCheck.xml: is filled by analyse part, and contains only problems encountered by services configuration analyse.
- CatalogCheck.xml: is filled by analyse part, and contains only problems encountered by catalog analyse.

These 3 files are used to generate the global report, and are used as input for resolution part.

The global report file is called **AnalyseResult_<Date-Hour>.xml**.

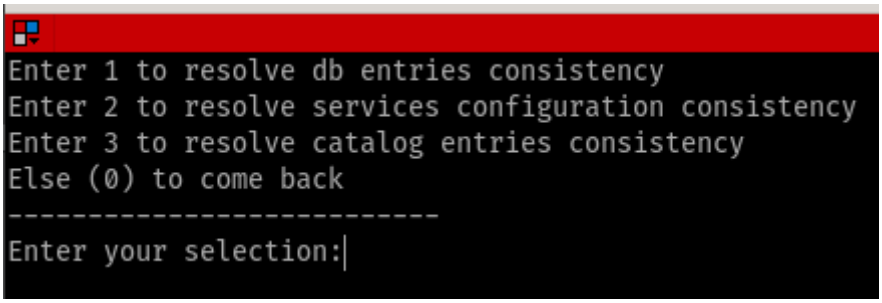
This analyse result can be send to DIATEAM support for analyse.

Press **0**, to come back to first menu, if you want to resolve problems. Otherwise you can close the menu.

2.2.3 Resolve data inconsistencies

For resolution part : Be sure nobody is working on CyberRange, when you want to solve any problem because the python script resolution does not communicate with hynesim. And then we cannot guaranty that the same data is not modified by correction process and by hynesim at same time.

From main menu, you can select 2, to enter into resolve menu.



You will see the following menu, that permit you to resolve datas from database, services configuration files, and catalog.

In this case, it is mandatory to resolve each part separately.

As mentioned below, the input element to permit resolution, are DBCheck.xml, ServicesConfigurationCheck.xml and CatalogCheck.xml files located into **/usr/share/hynesim/scripts/IntegrityChecker/analysisResult/** folder.

2.2.3.1 Database resolution

Element to resolve are located into **/usr/share/hynesim/scripts/IntegrityChecker/analysisResult/DBCheck.xml** file.

When entering into resolution mode, a backup of the following files is made:

- /etc/hynesim/catalog.db,
- /etc/hynesim/diamesh.db

They are just prefixed by current date.

- /etc/hynesim/catalog-ddMMYYYY-hhmmss.db,
- /etc/hynesim/diamesh-ddMMYYYY-hhmmss.db

The resolution action consists to execute sql command line, to delete obsolete datas detected by analyse.

For example, the analyze extract the minimum information, to build the sql request to send for resolution part. The orphan detected are element already present into junction table, where id/uuid were deleted by delete action. But if Master crash when deleting, we don't know if the database is aligned with file storage. Some element could be present, but never used. It is recommended to delete them.

For invalid ids, just the table name is requested, to build the delete request, to the corresponding database table.

For orphan elements :

- if analyze detect an orphan P/C link, we just need the key to delete from **catalog.db** file, **entity_relationship** table,
- if analyze detect an orphan group or user, we just need the key to delete from **diamesh.db** file, **join_groups_users** table.

For duplicate uuids, idem, we just need the duplicate uuid to delete from **catalog.db** file, **elements** table. We just keep only one declaration.

The resolution will send all deleted request to corresponding database table.

After made this correction, you must do a resetPlatform, if you are in 4.7 version, otherwise use restartPlatform, to permit hynesim-master and hynesim-node services, to resynchronize all datas with updated values.

If correction correctly applied, you can read on terminal, for each request:

Number of rows deleted = "Number" for "Table name".

Sometimes the number could be 0, because another delete action, previously made, already deleted 2 elements.

After that, you can relaunch the analyze, to be sure you detect less errors.

In case a problem encountered after resolving element, by script, don't forget you have a db files backup made automatically into /etc/hynesim folder.

To come back to previous situation, you just need to copy the backup file to official one.

- **cp /etc/hynesim/catalog-ddMMYYYY-hhmmss.db /etc/hynesim/catalog.db**
- **cp /etc/hynesim/diamesh-ddMMYYYY-hhmmss.db /etc/hynesim/diamesh.db**

After made this correction, you must do a resetPlatform, if you are in 4.7 version, otherwise use restartPlatform, to permit hynesim-master and hynesim-node services, to resynchronize all datas with original values.

DIATEAM - 31 rue Yves COLLET - 29200 Brest - FRANCE | Tel : +33(0) 298 050 050

2.2.3.2 Service configuration resolution

Element to resolve are located into **/usr/share/hynesim/scripts/IntegrityChecker/analysisResult/ServiceConfigurationCheck.xml** file.

When entering into resolution mode, a backup of the following files is made:

- /etc/hynesim/hynesim-glacier.conf,
- /etc/hynesim/hynesim-master.ini,
- /etc/hynesim/hynesim-node.ini

They are just prefixed by current date.

- /etc/hynesim/hynesim-glacier-ddMMYYYY-hhmmss.conf,
- /etc/hynesim/hynesim-master-ddMMYYYY-hhmmss.ini,
- /etc/hynesim/hynesim-node-ddMMYYYY-hhmmss.ini

As script is running on platform, it will duplicate the existing service configuration file with the date, and will regenerate a new one, with missing key/values.

But as we use Python library the new generated file, does not contains all comments.

All lines beginning by ";" or "#", will be deleted. So if explanation about parameters are present, the resolver will delete them.

You cannot do a difference between original and modified file.

If you want to keep comments, you can just edit the file on Range, and add the missing key/values.

For **/etc/hynesim/hynesim-glacier.conf** file, only **Ice.MessageSizeMax** key presence into [General] section and his value is checked.

Default value to check:

Ice.MessageSizeMax=131072

For **/etc/hynesim/hynesim-master.ini** file, only **SynchroPeriodInS** and **SynchroAtStartOnly** key presence into [Hynesim] section are checked.

After made this correction, you must do a resetPlatform, if you are in 4.7 version, otherwise use restartPlatform, to permit hynesim-glacier and hynesim-master services, to take in account the updated values.

After that, you can relaunch the analyze, to be sure you detect less errors.

In case a problem encountered after resolving element, by script, don't forget you have a service configuration files backup made automatically into /etc/hynesim folder.

To come back to previous situation, you just need to copy the backup file to official one.

- **cp /etc/hynesim/hynesim-glacier-ddMMYYYY-hhmmss.conf /etc/hynesim/hynesim-glacier.conf**
- **cp /etc/hynesim/hynesim-master-ddMMYYYY-hhmmss.ini /etc/hynesim/hynesim-master.ini**
- **cp /etc/hynesim/hynesim-node-ddMMYYYY-hhmmss.ini /etc/hynesim/hynesim-node.ini**

After made this correction, you must do a resetPlatform, if you are in 4.7 version, otherwise use restartPlatform, to permit hynesim-master and hynesim-node services, to resynchronize with original values.

2.2.3.3 Catalog configuration resolution

Element to resolve are located into **/usr/share/hynesim/scripts/IntegrityChecker/analysisResult/CatalogConfigurationCheck.xml** file.

In this case, you can encountered multiple problems:

- Orphan entities: Do nothing, it is just an information for admin, on possible entities to delete to free space on platform.
- Non domains templates into topologies,
- Entities declared into multiple topologies.

Each topology contains an XML file definition, where all entities (wires, domains, diodes, …) are defined. These declaration used XML attributes, used by hynesim-master, to load, unload topologies and content.

For non domains templates into topologies, one attribute called **isTemplate** is used to load entities. This attribute must be defined to 0 into topology definition. This resolution will replace all isTemplate=1 to isTemplate=0, into topology XML definition, to avoid multiple entity creation.

As we never encountered this case, for the moment the resolution just print the XML content after correction, but don't modify the original file. If encountered this case, if print content is correct, we can activate the code.

Multiple case can be found:

- The duplicated uuid is referenced only by wires shared by multiple topologies.
- The duplicated uuid is referenced by templates entities shared by multiple topologies.
- The duplicated uuid is referenced by single catalog entity with physical filestorage shared by multiple topologies.
- The duplicated uuid is referenced by single catalog domain with physical filestorage shared by multiple topologies.
- The duplicated uuid is referenced by different kind of entities/wires and shared by multiple topologies.

When resolving a topology, a backup of the following files is made:

- /data/hynesim/catalog/topologies/UUID/hnsEntryConfig.xml or /data/hynesim/catalog/topologies/import/UUID/hnsEntryConfig.xml

They are just prefixed by current date.

- /data/hynesim/catalog/topologies/UUID/hnsEntryConfig-ddMMYYYY-hhmmss.xml or /data/hynesim/catalog/topologies/import/UUID/hnsEntryConfig-ddMMYYYY-hhmmss.xml

For wire and template entity case, the resolver will replace the UUID associated to entity, with a new one not already used.

For non-domain entities with physical storage, we will replace entity registration by a topology template registration in the topology xml file, by extracting the xml definition of entity from its proper xml file. The UUID value now stored in inPlaceUUID property is set to a new unexisting value.

For domain with physical storage, we must inform the user, to let him make the correction, into listed topologies found by analyser.

For mixed case (never seen), the informations are given and an administrator will have to resolve the problems.

In case a problem encountered after resolving element, by script, don't forget you have a hnsEntryConfig.xml file backup made automatically into /data/hynesim/catalog/topologies/UUID or /data/hynesim/catalog/topologies/import/UUID folder.

To come back to previous situation, you just need to copy the backup file to official one.

- **cp /data/hynesim/catalog/topologies/UUID/hnsEntryConfig-ddMMYYYY-hhmmss.xml /data/hynesim/catalog/topologies/UUID/hnsEntryConfig.xml**

or

- **cp /data/hynesim/catalog/topologies/import/UUID/hnsEntryConfig-ddMMYYYY-hhmmss.xml /data/hynesim/catalog/topologies/import/UUID/hnsEntryConfig.xml**

In this case a restartPlatform is needed to take hnsEntryConfig.xml files updated in account.