Group-IB Threat Intelligence - Splunk integration.

Group-IB Splunk integration is performed via Python script that downloads data from Group-IB TI server and saves it as .csv files.
Splunk should be configured to monitor the directory where the files are saved.

1. Script configuration

Requirements:

1. Python 3.+
2. Python **requests** module.
3. Python **pyaml** module.

Installation:

1. Create directory for the script on your Splunk instance (for example, **%Splunk home directory%/group_ib/**) and copy all the provided files there ("gib_ti" folder, "configuration.yml" and "GIB_poller.py".

2. Open **configuration.yml** with some text editor and fill in **username** and **api_key** fields. **Username** is the login for Group-IB TI portal. The **api_key** can be manually generated in the portal (log in the portal → click on your name in right upper corner → choose **Profile** option → click on **Go to my setting** button under your name → under **Change password** button you will see **API KEY generator**. **Do not forget to save the API key**). Also you can configure proxy settings here in proxy section.

3. Inside **configuration.yml** you can find **collections** section. This section is used to enable collections and set initial date for each collection polling.

- To enable collection for uploading set enable field to **true**, to disable, set it to **false**.

- Also, you can set initial date in **YYYY-MM-DD** format. If it is not set, the intitial polling of each collection will start from 3 days ago date. After that you will get only actual data. (Do not use too big intreval between current date and initial date, or too much data will be uploaded, especially for suspicious_ip/* collections.
- Do not fill **seqUpdate** values. The script uses this field to get only actual data.

```
apt/threat:
        default_date: '2020-07-24'
        enable: true
        seqUpdate:
```

4. You can start script with the following command (specify your python version):

```
python3.7 api_poller.py
```

5. Script should be set to be executed every hour (this process described in Task scheduling).

2. Folder Monitoring

1. Log into Splunk. Click on **Settings** and choose **Data input** option → in **Local inputs** section click on **Files & Directories** and push on **New Local File & Directory** in right upper corner.

2. Input the directory where you put the script and add **/data** to it (for our example this will be **%Splunk home directory%/group_ib/data**).

3. Choose **Automatic** source type, desirable app context, **Constant value** as host, desirable index and click on **Review** button, then **Submit** button.

Now your data input is created and you can see it in the list.

3. Data model configuration

Here we will create Data model and fill it with data sets for all collections. Each collection should have separate dataset.

1. Click on **Settings** and choose **Data models** option, from the right upper corner choose **New Data Model** options, in the App field choose **Search & Reporting**.

2. Created data model will be opened, in the left part of the screen click on **Add Dataset** and choose **Root search**.

3. **Dataset Name** and **Dataset ID** could be filled with any data. For the search string use the following:

```
source="<directory from Part II>/<collection name>/*" index="<index>"
```

where **index** is the same that was chosen in Part II, step 3 (if **default** value was chosen then the index here should be **main**).

For example, for **Compromised account** collection, for directory from Part II and for **default** index the string will be following:

```
source="%<Splunk home directory>%/group_ib/data/compromised_account/*"
index="main"
```

This string is used to create dataset for **Compromised accounts** collection. For each other necessary collection a separate dataset should be created. The only difference between them and example is the **collection name** in the **source** value. Here is the full list of directory names for all available collections:

- apt_threat
- apt_threat_actor
- attacks_ddos
- attacks_deface
- attacks_phishing
- attacks_phishing_kit
- bp_phishing
- bp_phishing_kit
- compromised_account
- compromised_card
- compromised_file
- compromised_imei
- compromised_mule
- hi_threat
- hi_threat_actor
- malware_cnc

- malware_malware
- malware_targeted_malware
- osi_git_leak
- osi_public_leak
- osi_vulnerability
- suspicious_ip_open_proxy
- suspicious_ip_socks_proxy
- suspicious_ip_tor_node

There are also two additional directories for mitre_matrix data. This datasets will be awailable if **apt/threat** and **hi/threat** collections are enabled.

- apt_threat_mitre_matrix
- hi_threat_mitre_matrix

4. After dataset creation you would be able to add fields to each dataset. In the right part of the screen click on **Add Field** → **Auto-Extracted**. In case you had already downloaded data with script you will be suggested to add fields, otherwise you will have to add them manually (it can be done by **Add by Name** option). If in downloaded data some field wasn't met it won't be suggested automatically, so you should check all the fields (and its types) and add manually those that weren't added.

- compromised_account:
    - dateCompromised (String)
    - dateDetected (String)
    - client_ipv4_ip (IPV4)
    - login (String)
    - password (String)
    - domain (String)
    - cnc_ipv4_ip (IPV4)
    - cnc_cnc (String)

- compromised_card:
    - dateCompromised (String)
    - dateDetected (String)
    - cardInfo_number (String)
    - owner_name (String)
    - owner_city (String)
    - owner_email (String)
    - cnc_cnc (String)

    - cnc_ipv4_ip (IPV4)

- compromised_mule:

    - dateAdd (String)
    - dateIncident (String)
    - account (String)
    - cnc_cnc (String)
    - cnc_domain (String)
    - cnc_ipv4_ip (IPV4)
    - organization_name (String)
    - person_name (String)
    - person_email (String)
    - person_phone (String)
    - threatActor_name (String)

- compromised_imei:

    - dateDetected (String)
    - dateCompromised (String)
    - device_imei (String)
    - device_model (String)
    - cnc_cnc (String)
    - cnc_ipv4_ip (IPV4)

- compromised_file:

    - dateCompromised (String)
    - dateDetected (String)
    - fileInfo_filename (String)
    - fileInfo_md5 (String)
    - malware_name (String)
    - threatActor_name (String)
    - cnc_cnc (String)
    - cnc_ipv4_ip (IPV4)

- attacks_ddos:

    - dateReg (String)
    - dateBegin (String)
    - dateEnd (String)
    - target_ipv4_ip (IPV4)

    - target_domain (String)
    - target_url (String)
    - cnc_cnc (String)
    - cnc_ipv4_ip (IPV4)

- attacks_deface:

    - date (String)
    - targetDomain (String)
    - targetIp_ip (String)
    - threatActor_name (String)
    - url (String)

- attacks_phishing:

    - dateDetected (String)
    - dateBlocked (String)
    - ipv4_ip (IPV4)
    - phishingDomain_domain (String)
    - phishingDomain_dateRegistered (String)
    - phishingDomain_title (String)

- attacks_phishing_kit:

    - hash (String)
    - downloadedFrom_url (String)
    - downloadedFrom_domain (String)
    - emails (String)
    - targetBrand (String)

- bp_phishing:

    - dateDetected (String)
    - dateBlocked (String)
    - ipv4_ip (IPV4)
    - phishingDomain_domain (String)
    - phishingDomain_dateRegistered (String)
    - phishingDomain_title (String)

- bp_phishing_kit:

    - hash (String)

    - downloadedFrom_url (String)
    - downloadedFrom_domain (String)
    - emails (String)
    - targetBrand (String)

- hi_threat:

    - dateFirstSeen (String)
    - dateLastSeen (String)
    - datePublished (String)
    - threatActor_name (String)
    - indicators_params_ipv4 (IPV4)
    - indicators_params_hashes_md5 (String)
    - indicators_params_hashes_sha1 (String)
    - indicators_params_hashes_sha256 (String)

- hi_threat_actor:

    - createdAt (String)
    - name (String)
    - aliases (String, sep = “|”)
    - labels (String, sep = “|”)
    - langs (String, sep = “|”)

- apt_threat:

    - dateFirstSeen (String)
    - dateLastSeen (String)
    - datePublished (String)
    - threatActor_name (String)
    - indicators_params_ipv4 (IPV4)
    - indicators_params_hashes_md5 (String)
    - indicators_params_hashes_sha1 (String)
    - indicators_params_hashes_sha256 (String)

- apt_threat_actor:

    - createdAt (String)
    - name (String)
    - aliases (String, sep="|")
    - labels (String, sep="|")
    - langs (String, sep="|")

- osi_git_leak:
    - dateDetected (String)
    - name (String)
    - repository (String)
    - matchesType (String)
    - revisions_info_authorEmail (String)
    - revisions_info_authorName (String)
    - revisions_info_dateDetected (String)
- osi_public_leak:
    - datePublished (String)
    - created (String)
    - hash (String)
    - size (String)
    - matchType (String)
    - keyword (String)
    - link (String)
- osi_vulnerability:
    - datePublished (String)
    - id (String)
    - cvss_score (String)
    - description (String)
    - reporter (String)
    - software_softwareName (String)
    - software_softwareVersionString (String)
    - software_vendor (String)
- malware_cnc:
    - dateDetected (String)
    - dateLastSeen (String)
    - cnc (String)
    - ipv4_ip (IPV4)
    - domain (String)
    - threatActor_name (String)
    - malwareList_name (String)
    - platform (String)

- malware_malware:
    - name (String)
    - platform (String)
    - shortDescription (String)
    - threatLevel (String)
- malware_targeted_malware:
    - date (String)
    - malware_name (String)
    - md5 (String)
    - injectMd5 (String)
    - threatActor_name (String)
    - fileName (String)
    - fileType (String)
    - size (String)
    - source (String)
- suspicious_ip_tor_node:
    - dateFirstSeen (String)
    - dateLastSeen (String)
    - ipv4_ip (String)
    - source (String)
- suspicious_ip_open_proxy:
    - dateDetected (String)
    - dateFirstSeen (String)
    - ipv4_ip (IPV4)
    - type (String)
    - port (Number)
    - source (String)
- suspicious_ip_socks_proxy:
    - dateDetected (String)
    - dateFirstSeen (String)
    - ipv4_ip (IPV4)
    - source (String)

- apt_threat_mitre_matrix
    - threat_id (String)
    - threatActor_name (String)
    - title (String)
    - attackTactic (String)
    - attackType (String)
    - tacticId (String)
- hi_threat_mitre_matrix
    - threat_id (String)
    - threatActor_name (String)
    - title (String)
    - attackTactic (String)
    - attackType (String)
    - tacticId (String)

4. Task scheduling

To schedule script to be executed **cron** (for *nix-based systems) or **Task Scheduler** (for Windows) can be used.

1. Cron:

- Open file **/etc/crontab**
- Set the following job:

```
0 * * * * root cd %<Path to “data” directory>% &&
%<Path to python>%/python %<Path to script>%/api_poller.py
```

- This job will start polling each hour at 0 minutes

2. Task Scheduler:

- Create new file “**GIB_poller.bat**” with the following content:

```
“%<Path to python.exe>%” “%<Path to python script>%\api.poller”
 pause
```

- Go to **Control Panel** → Administrative Tools → Task Scheduler

- Choose **Create Task** option.
- Fill the name of the task and description.
- In **Triggers** tab create new daily trigger and set it to be repeated each hour for infinity amount of days.
- In **Actions** tab add new action and select .bat file created on the first step.
- Click **Ok** button. The Task is created.