

---

# **CyberCAPTOR-Client**

***Release 4.4.3***

February 10, 2016



<b>1</b>	<b>Table of Contents</b>	<b>3</b>
1.1	CyberCAPTOR-Client - Installation and Administration Manual . . . . .	3
1.1.1	Table of Contents . . . . .	3
1.1.2	Introduction . . . . .	3
1.1.3	Installation . . . . .	3
1.1.4	Administration . . . . .	5
1.1.5	Sanity check procedures . . . . .	5
1.1.6	Diagnosis Procedures . . . . .	6
1.2	CyberCAPTOR-Client - User and Programmer Guide . . . . .	6
1.2.1	Table of Contents . . . . .	6
1.2.2	Introduction . . . . .	7
1.2.3	User Guide . . . . .	7
1.2.4	Programmers Guide . . . . .	11



## FIWARE Cyber seCurity Attack graPh moniTORing - Client

This project is part of FIWARE. For more information, please consult [FIWARE website](#).

CyberCAPTOR is an implementation of the Cyber Security Generic Enabler, the future developments of the [Security Monitoring GE](#).

The last version of the documentation can be accessed online at <https://cybercaptor.readthedocs.org/projects/cybercaptor-client/en/latest/>.



---

## Table of Contents

---

### 1.1 CyberCAPTOR-Client - Installation and Administration Manual

This project is a part of FIWARE. For more information, please consult [FIWARE website] (<http://www.fiware.org/>).

CyberCAPTOR is an implementation of the Cyber Security Generic Enabler, the future developments of the [Security Monitoring GE] (<http://catalogue.fiware.org/enablers/security-monitoring>).

The high-level README file of CyberCAPTOR-Client can be found here.

#### 1.1.1 Table of Contents

- *Introduction*
- *Installation*
  - *Prerequisite*
  - *Installation*
  - *Test*
- *Administration*

#### 1.1.2 Introduction

This is the Installation and Administration Manual for CyberCAPTOR-Client.

#### 1.1.3 Installation

This part detailed the procedure to install correctly CyberCAPTOR-Client.

##### Prerequisite

CyberCAPTOR-Client has been tested with the following software, but it should be possible to launch it with any other HTTP server (Apache, nginx,...).

This installation procedure need :

- Ubuntu

- Python
- Chromium

### Installation from sources

1. Get sources from GitHub

```
git clone https://github.com/fiware-cybercaptor/cybercaptor-client.git
```

2. Run a HTTP server. For example, we use here Python's SimpleHTTPServer but any other HTTP server may be used.

Run SimpleHTTPServer to serve CyberCAPTOR-Client on port 8000:

```
cd cybercaptor-client
python -m SimpleHTTPServer 8000
```

Note that you need a CyberCAPTOR Server to test properly CyberCAPTOR-Client. CyberCAPTOR Server can be launched with Docker using this command :

```
docker run --name cybercaptor-server -p 8080:8080 fiwarecybercaptor/cybercaptor-server
```

More information about CyberCAPTOR-Server (can be found here)[<https://github.com/fiware-cybercaptor/cybercaptor-server/blob/master/README.md>].

### Installation with Docker

If you want to run the client in foreground in a terminal, launch the following command. CyberCAPTOR-Client will listen on port 8000.

```
docker run --rm --name cybercaptor-client -p 8000:80 fiwarecybercaptor/cybercaptor-client
```

Note that you need a CyberCAPTOR Server to test properly CyberCAPTOR-Client. CyberCAPTOR Server can be launched with Docker using this command :

```
docker run --name cybercaptor-server -p 8080:8080 fiwarecybercaptor/cybercaptor-server
```

More information about CyberCAPTOR-Server (can be found here)[<https://github.com/fiware-cybercaptor/cybercaptor-server/blob/master/README.md>].

More details about building and/or running the Docker container can be found in Docker README.md.

### Test

Open your browser, for example Chromium, and go on URL :

```
http://localhost:8000
```

If you see a window with the title : *CyberCAPTOR-Client* and a tab : *Initialization*. The CyberCAPTOR-Client has been properly installed.



## 1.1.4 Administration

### Configuration file

The configuration file of CyberCAPTOR-Client allows to change the URL of CyberCAPTOR-Server.

This file is located in `js/myApp.js`.

The URL can be customized in the following block :

```
myApp.constant("myConfig", {
  // URL base for REST request
  "url": "http://localhost:8080/cybercaptor-server/rest/json",
  "config" : "http://localhost:8080/cybercaptor-server/rest/json/configuration/remediation-cost-par
})
```

## 1.1.5 Sanity check procedures

### End to End testing

Open your browser, for example Chromium, and go on URL :

```
http://localhost:8000
```

If you see a window with the title : *CyberCAPTOR-Client* and a tab : *Initialization*. The CyberCAPTOR-Client has been properly installed.

### List of Running Processes

#### Execution via Python's HTTPSimpleServer

```
# Results of ps -aux
user          9856  0.2  0.1  40812 13052 pts/4    S+   11:42   0:00 python -m SimpleHTTPServer 8000
```

#### Execution via Docker

```
# Results of ps -aux in docker container
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.4  0.1  29332 10716 ?        Ss   09:40   0:00 /usr/bin/python3 -u /sbin/my_init
root          11  0.0  0.0    196    40 ?        S    09:40   0:00 /usr/bin/runsvdir -P /etc/service
root          12  0.0  0.0    176     4 ?        Ss   09:40   0:00 runsv nginx
root          13  0.0  0.0    176     4 ?        Ss   09:40   0:00 runsv nginx-log-forwarder
root          14  0.0  0.0    176     4 ?        Ss   09:40   0:00 runsv syslog-ng
root          15  0.0  0.0    176     4 ?        Ss   09:40   0:00 runsv sshd
root          16  0.0  0.0    176     4 ?        Ss   09:40   0:00 runsv cron
root          17  0.0  0.0    176     4 ?        Ss   09:40   0:00 runsv syslog-forwarder
root          18  0.0  0.0  26752  2676 ?        S    09:40   0:00 /usr/sbin/cron -f
root          19  0.0  0.0   7480   776 ?        S    09:40   0:00 tail -F -n 0 /var/log/syslog
root          21  0.0  0.1 140232 12400 ?        S    09:40   0:00 nginx: master process /usr/sbin/nginx
root          22  0.0  0.0   63676  6648 ?        S    09:40   0:00 syslog-ng -F -p /var/run/syslog-ng.p
root          31  0.0  0.1 446420  8904 ?        Ss1  09:40   0:00 Passenger watchdog
root          34  0.0  0.1 1080468 12212 ?        S1   09:40   0:00 Passenger core
nobody        45  0.0  0.1 315060 10180 ?        S1   09:40   0:00 Passenger ust-router
www-data      59  0.0  0.0 140564  6348 ?        S    09:40   0:00 nginx: worker process
```

root	68	0.0	0.0	7480	704	?	S	09:40	0:00	tail -F /var/log/nginx/error.log
root	69	0.0	0.0	18144	3256	?	Ss	09:40	0:00	bash
root	83	0.0	0.0	15572	2112	?	R+	09:41	0:00	ps -aux

## Network interfaces Up & Open

The only port that needs to be open is the one chosen either for Python's HTTPSimpleServer, either for Docker container. It is port 8000 in examples above.

## 1.1.6 Diagnosis Procedures

### Resource availability

The amount of RAM and hard disk needed for CyberCAPTOR-Client is very low for few simultaneous clients (generally the case for the use of this application). 128Mb of RAM and 100Mo of hard disk dedicated to the application should be enough.

### HTTP Server Log files

The logs of the HTTP server are directly printed in the Terminal for Python's HTTPSimpleServer. For Docker container, logs of the HTTP server can be displayed with such command :

```
docker exec cybercaptor-client tail -f /var/log/nginx/error.log /var/log/nginx/access.log
```

### Javascript console

The Javascript errors are displayed in the Javascript console of the web browser. For Chromium, such console can be accessed by pressing **Ctrl + Shift + I**.

## 1.2 CyberCAPTOR-Client - User and Programmer Guide

This project is a part of FIWARE. For more information, please consult [FIWARE website] (<http://www.fiware.org/>).

CyberCAPTOR is an implementation of the Cyber Security Generic Enabler, the future developments of the [Security Monitoring GE] (<http://catalogue.fiware.org/enablers/security-monitoring>).

The high-level README file of CyberCAPTOR-Client can be found [here](#).

### 1.2.1 Table of Contents

- *Introduction*
- *User Guide*
  - *CyberCAPTOR-Client views*
    - \* *Initialization*
    - \* *Configuration*
    - \* *Attack Graph*

- \* *Attack Path*
- \* *Remediation Simulation*
- \* *Dynamic Risk Analysis*
- *Interpretation*
- *Programmer Guide*
  - *Technologies*
    - \* *AngularJS*
    - \* *D3JS*
    - \* *Bootstrap*
  - *Source files organization*
    - \* *JS*
      - *MyApp*
      - *Controller*
      - *Directive*
      - *Service*
      - *Filter*
    - \* *Lib*
      - *Transform*
    - \* *View*
    - \* *Img*
    - \* *Doc*

## 1.2.2 Introduction

This is the User and Programmer Guide of CyberCAPTOR-Client.

For the illustrations of this manual, we used the file `dataSet.xml` as topological input file.

## 1.2.3 User Guide

This guide describe how to use CyberCAPTOR-Client.

### CyberCAPTOR-Client views

#### Initialization

This page can be accessed at this adress : <http://localhost:8000/#/welcome>

This page initializes the server with data provided in the topological XML file.

Use the button to select your topology file. When it is loaded on the queue file, click on “Upload All” to upload all your data in the server. When your data is loaded, the progress bar is fulfilled and a message appears to acknowledge the good reception.

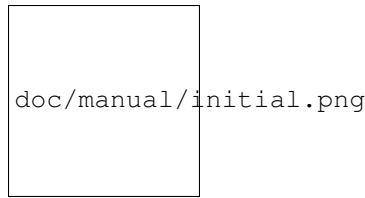


Fig. 1.1: Initialization page

Now, the server has received your data and CyberCAPTOR is ready for risk analysis.

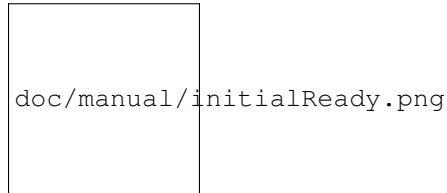


Fig. 1.2: Server ready

## Configuration

This page can be accessed at this adress : <http://localhost:8000/#/configuration>

This page lists all hosts of the network. You can also specify the importance of each host and update the parameters used for remediation cost calculation.

The panel “Configuration” lists all hosts of the network topology. You can filter this list with the input “Search”.

Click on the select input under “Name” to specify the importance of this host. By default, they host importance is “Negligeable”. When you are ready, click on “Save” to transmit the information to the server.

The other panel lists the parameters used for the remediation cost calculation. Change them according to your preferences and click on “Save”.

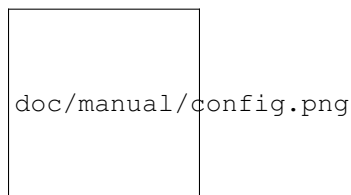


Fig. 1.3: Configuration page

## Attack Graph

This page can be accessed at this adress : <http://localhost:8000/#/attackGraph>

This page displays the attack graph of the information system.

By default, the graph is displayed in a topological view but, you can switch in a logical view by selecting the proper mode. If you put your cursor above a node, you can see the node details. You can also move the nodes using drag and drop.



Fig. 1.4: Attack Graph page

### Attack Path

This page can be accessed at this adress : <http://localhost:8000/#/attackPath>

This page displays the selected attack path, its attrition level and remediations.

You can select the path to view in the panel “Selection”. By default, the first path is displayed. The attrition level characterize the criticity of the path. There are five criticity level : Negligeable, Minor, Medium, Severe and Catastrophic.

By default, the graph is displayed in a topological view but, you can switch in a logical view by selecting the proper mode. If you put your cursor above a node, you can see the node details. You can also move the nodes using drag and drop.

Remediations lists all known solutions to correct the risk of the selected attack path. They are ordered by your habits (previously deployed remediations) and by the cost of the remediations. Habits represents your preference to a specific remediation.

The button “Simulate” open a new page “Remediation Simulation”.

You can see the attack path in a logical view.

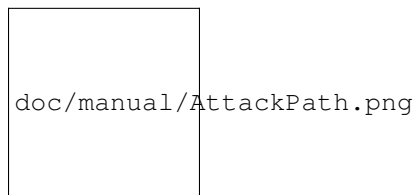


Fig. 1.5: Attack path, logical view

Or in a topological view

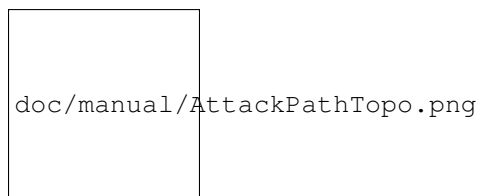


Fig. 1.6: Attack path, topological view

### Remediation Simulation

This page can be accessed at this adress : <http://localhost:8000/#/simulation>

This page display the simulation of a remediation on the whole attack graph.

In “Remediation Simulation”, you can see the attack graph. Nodes with a green border are corrected by the remediation selected and the nodes with orange border are still presents.

If you wan to confirm the remediation application, click on “Validate” to certify that you are going to apply this remediation. This action increments the habit score of this remediation.

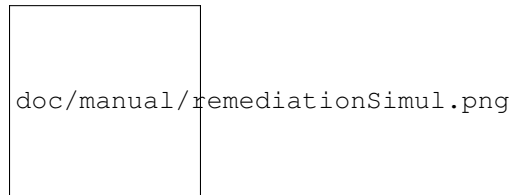


Fig. 1.7: Remediation Simulation page

### Dynamic Risk Analysis

This page can be accessed at this adress : <http://localhost:8000/#/dynamicRiskAnalysis>

This page allow to visualize the currently happening attacks on your information system received by the server in IDMEF.

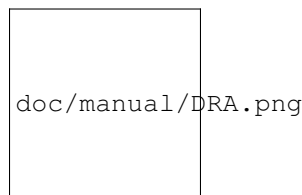


Fig. 1.8: Dynamic Risk Analysis page

The alarms are stored in the Alarm Box, you can selected one and see its impact on the whole information system.

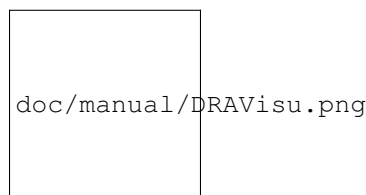


Fig. 1.9: Dynamic Risk Analysis visualization

To see the dynamic remediations known to solve the vulnerability, click on “Remediations”.

### Interpretation

This part explains how can understood the logical graphs displayed by CyberCAPTOR-Client.

In the following example, there are 5 nodes :

- node (1) : Physical access

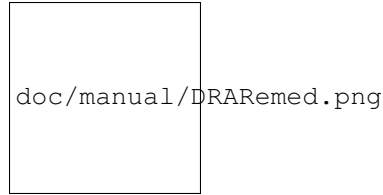


Fig. 1.10: Dynamic Risk Analysis remediations

- node (2) : Network access
- node (3) : Vulnerability
- node (4) : Rule for remote exploit
- node (5) : Execute code on “linux-user 2” as user

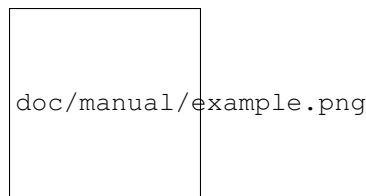


Fig. 1.11: Interpretation

The target, “linux-user-2”, has a network access and a physical access, a vulnerability is presents and these conditions allow an attacker to use a remote exploit. In this case, the attack can execute a code on the device as a user.

## 1.2.4 Programmers Guide

This guide describe how to develop within CyberCAPTOR-Client.

### Technologies

This part lists all technologies used to develop CyberCAPTOR-Client.

#### AngularJS

The Javascript framework [AngularJS](#) is used. You can find the documentation [here](#).

The library [Angular-File-Upload](#) is used to upload file.

#### D3JS

The Javascript library Data-Driven Documents [D3JS](#) is used to display the graphs.

#### Bootstrap

The framework [Bootstrap](#) is used to design CyberCAPTOR-Client (CSS + Javascript).

### Source files organization

This part presents the organization of the sources files, and the role of each folder.

#### JS

This section detailed all JavaScript files contains in the `js` folder.

**MyApp** This file contains all parameters, routes, constants of CyberCAPTOR-Client.

**Controller** This file contains all [controllers](#) used to manage CyberCAPTOR-Client.

**Directive** This file contains all [directives](#) used to display all graphe in CyberCAPTOR-Client.

**Service** This file contains all [services](#) used in CyberCAPTOR-Client.

**Filter** This file contains all [filters](#) used in CyberCAPTOR-Client.

#### Lib

**Transform** Transform owns differents methods used to modify data's structure received from server. That allow to simplify the calculations and the visualizations for these graphs.

#### View

This folder contains all views used to display informations, graphes, data,...

#### Img

This folder contains all pictures used in CyberCAPTOR Client.

#### Doc

This folder contains all documents describing CyberCAPTOR.

Information about development is also available [in the README file](#).

Please see the [project license](#) for license information.