CyberCAPTOR-Server

Release 4.4.3

September 29, 2015

Contents

1 Table of Contents **3**

1.1 CyberCAPTOR-Server - Installation and Administration Manual . . . 3

1.1.1 Table of Contents . . . 3

1.1.2 Introduction . . . 3

1.1.3 Installation . . . 3

Development Version Installation . . . 3

Prerequisite . . . 3

Build . . . 4

Installation . . . 4

Docker Version Deployment . . . 4

Build container (optional) . . . 4

Run container . . . 4

Test . . . 5

1.1.4 Administration . . . 5

Configuration file . . . 5

1.1.5 Sanity check procedures . . . 5

End to End testing . . . 5

List of Running Processes . . . 6

Execution of .war with tomcat7 . . . 6

Execution via Docker . . . 6

Network interfaces Up & Open . . . 6

1.1.6 Diagnosis Procedures . . . 7

Resource availability . . . 7

Main logs files . . . 7

1.2 CyberCAPTOR-Server - User and Programmer Guide . . . 7

1.2.1 Table of Contents . . . 7

1.2.2 Introduction . . . 8

1.2.3 User Guide . . . 8

CyberCAPTOR-Server API . . . 8

API usage . . . 8

1.2.4 Programmer Guide . . . 9

Javadoc . . . 9

API verification . . . 9

FIWARE Cyber seCurity Attack graPh moniTORing - Server

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.

This documentation can be accessed online at https://cybercaptor.readthedocs.org/projects/cybercaptor-server/en/latest/.

CHAPTER 1

Table of Contents

1.1 CyberCAPTOR-Server - 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-Server 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-Server.

1.1.3 Installation

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

Development Version Installation

Prerequisite

CyberCAPTOR-Server has been tested with the following software, but it should be possible to build and run in on all *Linux* OS, with Java 7.

- Ubuntu

- Java 1.7
- Apache Tomcat 7
- Apache Maven 3
- XSB
- MulVAL

Build

1. Get sources from Github

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

2. Use Maven to download dependencies and build the web application archive (.war).

```
mvn clean
mvn package
```

Installation

1. Deploy the .war into tomcat.

Using command line

```
cp ./target/cybercaptor-server*.war /var/lib/tomcat7/webapps/cybercaptor-server.war
```

This can also be done using the tomcat GUI manager, or with Maven's tomcat7 plugin.

2. Link the configuration and scripts repertory and fix permissions

```
sudo ln -s `pwd`/configuration-files /usr/share/tomcat7/.remediation
sudo ln -s `pwd`/src/main/python/ /usr/share/tomcat7/python_scripts
chmod -R o+rw ./configuration-files/
sudo chown -R tomcat7:tomcat7 /usr/share/tomcat7/
```

3. Copy and edit the configuration file

```
cp ./configuration-files/config.properties.sample ./configuration-files/config.properties
vim ./configuration-files/config.properties
```

See in [#configuration] to see the description of all parameters used in the configuration file.

Docker Version Deployment

Build container (optional)

```
docker build -t cybercaptor-server .
```

Run container

If you want to run the server in foreground, launch the following command:

```
docker run --rm --name cybercaptor-server -p 8000:8080 fiwarecybercaptor/cybercaptor-server
```

If you want to run the server in background, launch the following command:

```
docker run -d --rm --name cybercaptor-server -p 8000:8080 fiwarecybercaptor/cybercaptor-server
```

Then, the application can be accessed at http://localhost:8000/cybercaptor-server/.

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

Test

Go on URL : http://localhost:8080/cybercaptor-server/rest/json/initialize

If the result is `{"status":"Loaded"}`, the application has been properly built and installed.

1.1.4 Administration

Configuration file

The configuration file of CyberCAPTOR-Server allows to select many parameters and file paths used by CyberCAPTOR-Server.

This file is located in `configuration-files/config.properties`.

```
xsb-path=/opt/XSB/bin # The XSB installation binary path
output-path=/root/.remediation/tmp # The output folder for temporary computations
mulval-path=/opt/mulval/ # MulVAL installation path
mulval-rules-path=/root/.remediation/rules-with-topology.P # The MulVAL rules description file
cost-parameters-path=/root/.remediation/cost-parameters # The folder in which the remediation cost pa
database-path=/root/.remediation/vulnerability-remediation-database.db # The path toward the remediat
python-path=/usr/bin/python # Python path
mulval-input-script-folder=/root/cyber-data-extraction/ # The folder in which the mulval input script
host-interfaces-path=/root/.remediation/inputs/hosts-interfaces.csv # The path where the CSV host int
vlans-path=/root/.remediation/inputs/vlans.csv # The path where the CSV vlans file is described (if u
routing-path=/root/.remediation/inputs/routing.csv # The path where the routing file is described (if
flow-matrix-path=/root/.remediation/inputs/flow-matrix.csv # The path where the CSV flow matrix file
vulnerability-scan-path=/root/.remediation/inputs/scan.nessus # The path where the Nessus XML file is
mulval-input=/root/.remediation/tmp/mulval-input-generated.P # The path where the MulVAL input file
topology-path=/root/.remediation/inputs/topology-generated.xml # The path where the topology file wil
remediations-history-path=/root/.remediation/remediations-history.bin # The path where the remediatio
alerts-temporary-path=/root/.remediation/alerts-temp.bin # The path where the IDMEF alerts are tempor
```

More information about the parameters can be found in CyberCAPTOR-Data-Extraction README.

1.1.5 Sanity check procedures

End to End testing

Go on URL : http://localhost:8080/cybercaptor-server/rest/json/initialize

If the result is `{"status":"Loaded"}`, the application has been properly built and installed.

List of Running Processes

Execution of .war with tomcat7

```
# Results of ps -aux
root         20 12.1  4.1 3753696 337544 ?      Sl   11:45   0:09 /usr/bin/java -Djava.util.logging.co
root         66  0.0  0.0   4448  1568 ?        S    11:46   0:00 /bin/sh /opt/mulval//utils/graph_ger
root        127  0.0  0.1  30076 14196 ?        R    11:46   0:00 /opt/XSB/config/x86_64-unknown-linux
```

Execution via Docker

When idle

```
# Results of ps -aux in docker container
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.4  0.1  28236  9584 ?        Ss   11:45   0:00 /usr/bin/python3 -u /sbin/my_init
root          8  0.0  0.0    196    40 ?        S    11:45   0:00 /usr/bin/runsvdir -P /etc/service
root          9  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv tomcat7
root         10  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv syslog-ng
root         11  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv sshd
root         12  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv cron
root         13  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv syslog-forwarder
root         14  0.0  0.0  26752  2688 ?        S    11:45   0:00 /usr/sbin/cron -f
root         15  0.0  0.0   7480   704 ?        S    11:45   0:00 tail -f -n 0 /var/log/syslog
root         16  0.1  0.0  65760  6672 ?        S    11:45   0:00 syslog-ng -F -p /var/run/syslog-ng.p
root         17  0.0  0.0  21088  3196 ?        S    11:45   0:00 bash ./run
root         20 60.5  4.0 3749936 329468 ?      Sl   11:45   0:09 /usr/bin/java -Djava.util.logging.co
```

When MuLlVAL is running

```
# Results of ps -aux in docker container
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.1  28236  9584 ?        Ss   11:45   0:00 /usr/bin/python3 -u /sbin/my_init
root          8  0.0  0.0    196    40 ?        S    11:45   0:00 /usr/bin/runsvdir -P /etc/service
root          9  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv tomcat7
root         10  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv syslog-ng
root         11  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv sshd
root         12  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv cron
root         13  0.0  0.0    176     4 ?        Ss   11:45   0:00 runsv syslog-forwarder
root         14  0.0  0.0  26752  2688 ?        S    11:45   0:00 /usr/sbin/cron -f
root         15  0.0  0.0   7480   704 ?        S    11:45   0:00 tail -f -n 0 /var/log/syslog
root         16  0.0  0.0  65760  6672 ?        S    11:45   0:00 syslog-ng -F -p /var/run/syslog-ng.p
root         17  0.0  0.0  21088  3196 ?        S    11:45   0:00 bash ./run
root         20 12.1  4.1 3753696 337544 ?      Sl   11:45   0:09 /usr/bin/java -Djava.util.logging.co
root         66  0.0  0.0   4448  1568 ?        S    11:46   0:00 /bin/sh /opt/mulval//utils/graph_ger
root        127  0.0  0.1  30076 14196 ?        R    11:46   0:00 /opt/XSB/config/x86_64-unknown-linux
```

Network interfaces Up & Open

The only port that needs to be open is the one chosen either by tomcat server, either for Docker container. It is port 8080 in examples above.

1.1.6 Diagnosis Procedures

Resource availability

The amount of RAM and hard disk needed for CyberCAPTOR-Server can be high, according to the network topology. 8Gb of RAM and 1Go of hard disk dedicated to the application should be enough for a small-medium systems. For medium to big information systems, 32Gb of RAM and 30Go of hard disk dedicated to the application may be needed.

Main logs files

The main logs of the application can be accessed with

- `/var/log/tomcat7/catalina.out`
- `` `pwd`/configuration-files/tmp/xsb_log.txt ``
- `` `pwd`/configuration-files/tmp/input-generation.log ``

In docker container, they can be accessed with the following commands:

- `docker exec cybercaptor-server tail -n 50 -f /var/log/tomcat7/catalina.out`
- `docker exec cybercaptor-server tail -f /root/.remediation/tmp/tmp/xsb_log.txt`
- `docker exec cybercaptor-server tail -f /root/.remediation/tmp/tmp/input-generation.log`

1.2 CyberCAPTOR-Server - 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-Server can be found here.

1.2.1 Table of Contents

- *Introduction*
- *User Guide*
  - *CyberCAPTOR-Server API*
    - *API usage*
    - *Version API calls*
    - *Initialization calls*
      - *Attack graph, attack paths and remediation calls*
- *Programmer Guide*
  - *Javadoc*
  - *API verification*

1.2.2 Introduction

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

1.2.3 User Guide

This guide describe how to use CyberCAPTOR-Server.

CyberCAPTOR-Server API

CyberCAPTOR-Server only contains the REST API Server of CyberCAPTOR. Thus, it can be used only via its REST API. If you want a GUI for CyberCAPTOR-Server, you can use CyberCAPTOR-Client which is described in [https://github.com/fiware-cybercaptor/cybercaptor-client].

API usage

Version API calls To use the CyberCAPTOR server API, the first call to test that the server is available is

```
curl http://localhost:8080/cybercaptor-server/rest/version/detailed
```

which should returns something like

```
{"version":"4.4"}
```

Initialization calls Before using the API to manipulate the attack graph, the attack paths, and the remediations, the first call that needs to be done is

```
curl -c /tmp/curl.cookie http://localhost:8080/cybercaptor-server/rest/json/initialize
```

which loads the topology, generates the attack graph with MulVAL and computes the attack paths.

Note the `-c /tmp/curl.cookie` option of curl, allowing to keep the session cookie, necessary to chain calls and keep the attack graph and attack paths in session.

It is also possible to load the topology from an XML file, or a XML string containing the XML network topology, using the POST method of the `/rest/json/initialize` call :

Using a XML String:

```
curl -c /tmp/curl.cookie -H "Content-Type: application/xml" -X POST -d '<topology><machine><name><linu
```

Using a XML file:

```
curl -c /tmp/curl.cookie -X POST  -H "Content-Type: multipart/form-data"  -F "file=@./topology.xml" h
```

The exhaustive description of this file is XML topological file is provided in https://github.com/fiware-cybercaptor/cybercaptor-data-extraction/blob/master/doc/topology-file-specifications.md. This file can be generated automatically using CyberCAPTOR-Data-Extraction.

Attack graph, attack paths and remediation calls Then, the calls to get the attack paths, attack graph or remediations can be used:

Get the number of attack paths:

```
curl -b /tmp/curl.cookie http://localhost:8080/cybercaptor-server/rest/json/attack_path/number
```

Note the `-b /tmp/curl.cookie` option of curl, to load the previously saved session cookie.

Get the attack path 0:

```
curl -b /tmp/curl.cookie http://localhost:8080/cybercaptor-server/rest/json/attack_path/0
```

Get the attack graph

```
curl -b /tmp/curl.cookie http://localhost:8080/cybercaptor-server/rest/json/attack_graph
```

Get the remediations for attack path 0:

```
curl -b /tmp/curl.cookie http://localhost:8080/cybercaptor-server/rest/json/attack_path/0/remediation
```

Get the XML network topology (useful for backups):

```
curl -b /tmp/curl.cookie http://localhost:8080/cybercaptor-server/rest/json/topology
```

The full list of API calls and specifications is stored in apiary.apib and can be visualized on Apiary.io using the Apiary Blueprint format.

1.2.4 Programmer Guide

This guide describe how to develop within CyberCAPTOR-Server.

Javadoc

The Javadoc of CyberCAPTOR-Server as well as many interesting information for developers can be found on github pages: Developer pages - Javadoc.

Javadoc can be updated directly with Maven using

```
mvn site-deploy
```

Don't forget to configure GitHub OAuth token in `~/.m2/settings.xml`. Tokens can be generated on https://github.com/settings/tokens, with repo and user:email authorized scopes.

```
<settings>
      <servers>
          <server>
                <id>github</id>
                <password>OAuth token</password>
          </server>
      </servers>
</settings>
```

API verification

The API specified using Blueprint can be checked with the dredd tool. In order to do that, first install bredd with NPM (you should have Node.js installed).

```
sudo npm install -g dredd
```

Go in the folder in which is the dredd configuration file tools/api/dredd.yml:

```
cd tools/api
```

Execute dredd

```
dredd
```

In addition to the console reports provided by dredd, a detailed report file can be found in `tools/api/report.html`.

Information about development is also available in the README file.

Please see the project license for license information.