

# User Manual

# Contents

<b>1</b>	<b>Installation</b>	<b>2</b>
1.1	Overview . . . . .	2
1.2	Openvas . . . . .	3
1.3	Dependency-check . . . . .	4
1.4	Webissues . . . . .	4
<b>2</b>	<b>Use</b>	<b>5</b>
2.1	add a project . . . . .	5
2.2	add a member . . . . .	6
2.3	add a server . . . . .	6
2.4	add a code . . . . .	7
2.5	scan the targets . . . . .	7
	2.5.1 Dynamic scan with openvas . . . . .	7
	2.5.2 Static scan with dependency-check . . . . .	8
2.6	Results . . . . .	8
<b>3</b>	<b>Jobs</b>	<b>9</b>

# Chapter 1

## Installation

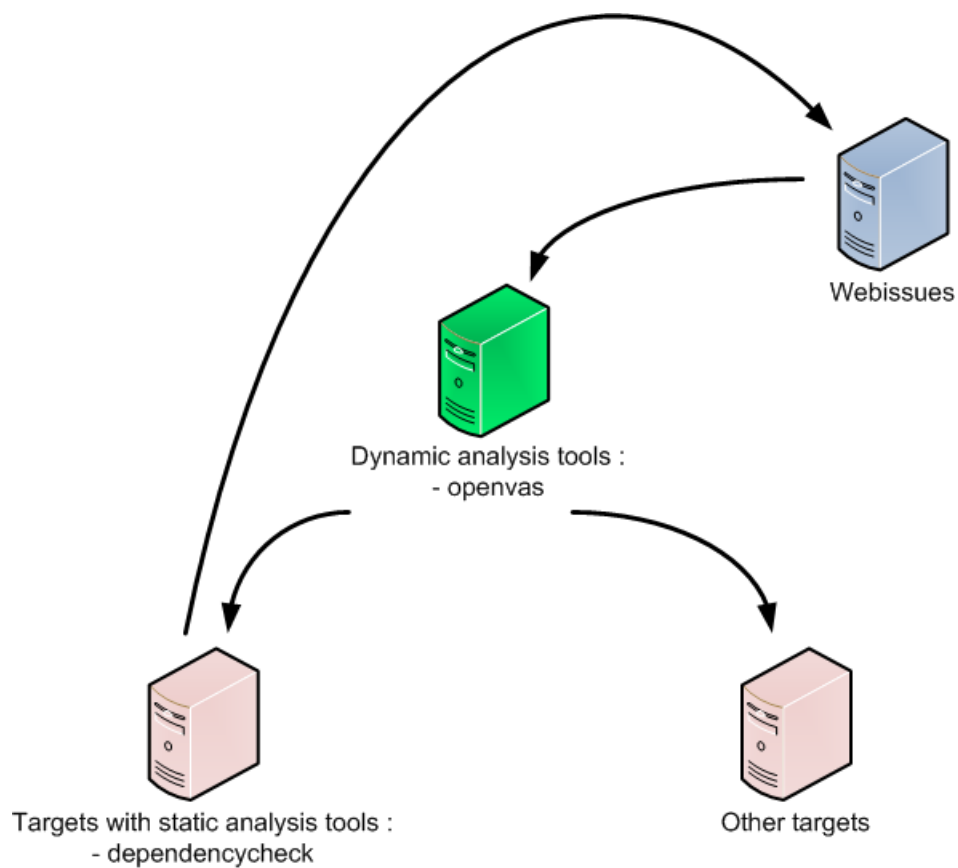
### 1.1 Overview

*Security-bugtracker* is currently a tool based on three dependencies :

- webissues : a bug tracker  
<http://webissues.mimec.org/>
- openvas : a dynamic security vulnerabilities assessment tool  
<http://www.openvas.org/>
- dependency-check : a static security vulnerabilities assessment tool  
<https://github.com/jeremylong/DependencyCheck>

Each of this tool can be installed on different or same server.

The aim of this project is to produce automated security tests and track detected default in a bugtracker.



## 1.2 Openvas

See the documentation on the official web site : <http://www.openvas.org/install-source.html>

On the same server install a web server and php, then copy the following module of this project to the directory of your web server :/security-bugtracker/security\_tools/openvas

Then edit /security-bugtracker/security\_tools/openvas/openvas.conf.php :

```
1 <?php
2
3 $CONF_WS_OPENVAS_LOGIN = "test";
4 $CONF_WS_OPENVAS_PASSWORD = "test";
5 $CONF_WEBISSUES_OPENVAS_LOGIN = "openvas";
6 $CONF_WEBISSUES_OPENVAS_PASSWORD = "openvas";
7 $CONF_WEBISSUES_WS_ENDPOINT = "http://localhost:8080/webissues-server-1.1.4/client/
  webservices.php";
8 $CONF_OPENVAS_ALERT_URL = "http://localhost:8080/webissues-server-1.1.4/client/
  security_tools/openvas/openvas.php";
9 $CONF_OPENVAS_ADMIN_LOGIN = "admin";
10 $CONF_OPENVAS_ADMIN_PASSWORD = "0825839c-0d3f-4417-a118-954a78e2553c";
11 $CONF_OPENVAS_CONFIG_ID = "a0e8fed8-45c1-4890-bd08-671257f63308";
12 $CONF_OPENVAS_PATH_OMP = "/usr/local/bin/omp";
13 $CONF_OPENVAS_PORT_OMP = "9393";
14
15 ?>
```

- CONF\_WS\_OPENVAS\_LOGIN
- CONF\_WS\_OPENVAS\_PASSWORD

are the credentials for the web services of this module.

- CONF\_WEBISSUES\_OPENVAS\_LOGIN
- CONF\_WEBISSUES\_OPENVAS\_PASSWORD
- CONF\_WEBISSUES\_WS\_ENDPOINT

will be completed later.

- CONF\_OPENVAS\_ALERT\_URL

is the address of this module on this web server.

- CONF\_OPENVAS\_ADMIN\_LOGIN
- CONF\_OPENVAS\_ADMIN\_PASSWORD

are the openvas admin credentials.

- CONF\_OPENVAS\_CONFIG\_ID

is the default config id for run a scan with openvas, check your config with this openvas command

```
linux-3ig5:/home/eric/security-bugtracker/documentation # omp -u admin -w 0825839c-0d3f-4417-a118-954
  a78e2553c -p 9393 --get-configs
8715c877-47a0-438d-98a3-27c7a6ab2196 Discovery
085569ce-73ed-11df-83c3-002264764cea empty
daba56c8-73ec-11df-a475-002264764cea Full and fast
698f691e-7489-11df-9d8c-002264764cea Full and fast ultimate
708f25c4-7489-11df-8094-002264764cea Full and very deep
a0e8fed8-45c1-4890-bd08-671257f63308 Full and very deep Clone 1
74db13d6-7489-11df-91b9-002264764cea Full and very deep ultimate
2d3f051c-55ba-11e3-bf43-406186ea4fc5 Host Discovery
bbca7412-a950-11e3-9109-406186ea4fc5 System Discovery
```

- CONF\_OPENVAS\_PATH\_OMP

is the path of your omp binary on this server.

- CONF\_OPENVAS\_PORT\_OMP

is the tcp port which on openvas / omp is running

## 1.3 Dependency-check

See the documentation on the official web site : <https://github.com/jeremylong/DependencyCheck>

## 1.4 Webissues

See the documentation on the official web site : <http://wiki.mimec.org/wiki/WebIssues/Installation>. Once the bugtracker is installed, copy the following module of this project to your webissues root directory :  
`/security-bugracker/webissues-server-1.1.4`

Next go at this address (replace the name, port, path with rights informations) :  
`http://localhost:8080/webissues-server-1.1.4/client/securityplugin.php`



Select *install plugin* and enter choosen values when the openvas module was installed above :

- CONF\_WS\_OPENVAS\_LOGIN
- CONF\_WS\_OPENVAS\_PASSWORD
- CONF\_OPENVAS\_ALERT\_URL

For finish create *openvas* and *Dependency-check* users in webissues.

# Chapter 2

## Use

Don't forget to use *basic authentication* with a login which have the good rights on webissues when using the webservices.

### 2.1 add a project

Add a project with the following web service method or via the traditional him of web issues :

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:v1="http
2   ://securitybugtracker/V1">
3   <soapenv:Header/>
4   <soapenv:Body>
5     <v1:addproject>
6       <name>TEST</name>
7       <description>TEST</description>
8     </v1:addproject>
9   </soapenv:Body>
10 </soapenv:Envelope>
```

Remember the ids returned with the response :

```
1 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="
2   http://securitybugtracker/V1">
3   <SOAP-ENV:Body>
4     <ns1:addproject_Response>
5       <id_details>
6         <id_project>29</id_project>
7         <id_folder_bugs>81</id_folder_bugs>
8         <id_folder_servers>82</id_folder_servers>
9         <id_folder_codes>83</id_folder_codes>
10        <id_folder_scans>84</id_folder_scans>
11      </id_details>
12    </ns1:addproject_Response>
13  </SOAP-ENV:Body>
14 </SOAP-ENV:Envelope>
```

## 2.2 add a member

Add a *robot* member for this project (the *openvas* account created during the installation) :

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:v1="http
  ://securitybugtracker/V1">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <v1:addmember>
5       <id_user>4</id_user>
6       <id_project>29</id_project>
7       <access>admin</access>
8     </v1:addmember>
9   </soapenv:Body>
10 </soapenv:Envelope>
```

```
1 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="
  http://securitybugtracker/V1">
2   <SOAP-ENV:Body>
3     <ns1:addmember_Response>
4       <result_details>
5         <result>true</result>
6       </result_details>
7     </ns1:addmember_Response>
8   </SOAP-ENV:Body>
9 </SOAP-ENV:Envelope>
```

## 2.3 add a server

Add a target server for this project, you can add multiple ips separated by the , character and the values of *use parameter* must be one of thoses :

- Development : for a development environment server
- Test : for a test environment server
- Production : for a production environment server

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:v1="http
  ://securitybugtracker/V1">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <v1:addserver>
5       <id_folder_servers>82</id_folder_servers>
6       <hostname>eric-pc</hostname>
7       <description>eric-pc</description>
8       <use>Production</use>
9       <ipsaddress>127.0.0.1</ipsaddress>
10    </v1:addserver>
11  </soapenv:Body>
12 </soapenv:Envelope>
```

```
1 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="
  http://securitybugtracker/V1">
2   <SOAP-ENV:Body>
3     <ns1:addserver_Response>
4       <result_addserver_details>
5         <id_server>1676</id_server>
6       </result_addserver_details>
7     </ns1:addserver_Response>
8   </SOAP-ENV:Body>
9 </SOAP-ENV:Envelope>
```

## 2.4 add a code

Add a target code path for this project, the *code parameter* is the path of the directory which contain librairies to be scanned by the dependency-check security tool.

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:v1="http
  ://securitybugtracker/V1">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <v1:addcode>
5       <id_folder_codes>83</id_folder_codes>
6       <name>java test</name>
7       <description>java tes</description>
8       <code>/home/eric/test/libs-java</code>
9     </v1:addcode>
10  </soapenv:Body>
11 </soapenv:Envelope>
```

```
1 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="
  http://securitybugtracker/V1">
2   <SOAP-ENV:Body>
3     <ns1:addcode_Response>
4       <result_addcode_details>
5         <id_code>1680</id_code>
6       </result_addcode_details>
7     </ns1:addcode_Response>
8   </SOAP-ENV:Body>
9 </SOAP-ENV:Envelope>
```

## 2.5 scan the targets

### 2.5.1 Dynamic scan with openvas

Run a scan with openvas security tool, select openvas value for the *tool parameter*, select a specific openvas config scan if you don't want to use the default config parametered during the installation and select a filter which can be :

- info : only add issues with a severity equal or upper to info
- minor : only add issues with a severity equal or upper to minor
- medium : only add issues with a severity equal or upper to medium
- high : only add issues with a severity equal to high

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:v1="http
  ://securitybugtracker/V1">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <v1:addscan>
5       <id_folder_scans>88</id_folder_scans>
6       <name>test scan soap ui</name>
7       <description>test scan soap ui</description>
8       <tool>openvas</tool>
9       <filter>medium</filter>
10      <!--Optional:-->
11      <id_config_openvas>?</id_config_openvas>
12    </v1:addscan>
13  </soapenv:Body>
14 </soapenv:Envelope>
```



```

1 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="
2   http://securitybugtracker/V1">
3   <SOAP-ENV:Body>
4     <ns1:addscan_Response>
5       <result_addscan_details>
6         <id_scan>2422</id_scan>
7       </result_addscan_details>
8     </ns1:addscan_Response>
9   </SOAP-ENV:Body>
10 </SOAP-ENV:Envelope>

```

## 2.5.2 Static scan with dependency-check

The static scan must be run locally, see jobs chapter.

## 2.6 Results

You can view the results of your precedings actions with the help of webissues :

**Projets** [Gérer les projets](#)

Nom	Type
Tous les projets	
TEST	
Bugs	Bugs
Codes	Codes
Scans	Scans
Servers	Servers

**Servers**
[Ajouter une demande](#) | [Tout marquer comme lue](#) | [Tout marquer comme non lue](#) | [Gérer les vues](#) | [Gérer les alertes](#) | [Exporter en CSV](#)
  
Vue : Toutes les demandes | [Ajouter une vue](#)
  

ID	Nom	ips address	use
#1676	eric-pc	127.0.0.1	Production

**Projets** [Gérer les projets](#)

Nom	Type
Tous les projets	
TEST	
Bugs	Bugs
Codes	Codes
Scans	Scans
Servers	Servers

**Codes**
[Ajouter une demande](#) | [Tout marquer comme lue](#) | [Tout marquer comme non lue](#) | [Gérer les vues](#) | [Gérer les alertes](#) | [Exporter en CSV](#)
  
Vue : Toutes les demandes | [Ajouter une vue](#)
  

ID	Nom	code
#1680	java test	/home/eric/test/libs-java

**Projets** [Gérer les projets](#)

Nom	Type
Tous les projets	
TEST	
Bugs	Bugs
Codes	Codes
Scans	Scans
Servers	Servers

**Scans**
[Ajouter une demande](#) | [Tout marquer comme lue](#) | [Tout marquer comme non lue](#) | [Gérer les vues](#) | [Gérer les alertes](#) | [Exporter en CSV](#)
  
Vue : Toutes les demandes | [Ajouter une vue](#)
  

ID	Nom	Date de création	severity	time	tool
#1707	test	14/12/2015 06:22	medium	in progress	openvas

**Projets** [Gérer les projets](#)

Nom	Type
Tous les projets	
TEST	
Bugs	Bugs
Codes	Codes
Scans	Scans
Servers	Servers

**Bugs**
[Ajouter une demande](#) | [Tout marquer comme lue](#) | [Tout marquer comme non lue](#) | [Gérer les vues](#) | [Gérer les alertes](#) | [Exporter en CSV](#)
  
Vue : Toutes les demandes | [Ajouter une vue](#)
  

ID	Nom	Date de modification	Modifié par	Assigné à	État	Sévérité
#1717	MySQL Authentication Error Message User Enumeration Vulne...	14/12/2015 07:04	openvas		Actif	2
#1721	Check for SSL Weak Ciphers	14/12/2015 07:04	openvas		Actif	2
#1725	Deprecated SSLv2 and SSLv3 Protocol Detection	14/12/2015 07:04	openvas		Actif	2
#1729	DCShop exposes sensitive files	14/12/2015 07:04	openvas		Actif	2
#1733	http TRACE XSS attack	14/12/2015 07:04	openvas		Actif	2
#1737	MySQL weak password	14/12/2015 07:04	openvas		Actif	3
#1741	Proxy accepts CONNECT requests to itself	14/12/2015 07:04	openvas		Actif	3

# Chapter 3

## Jobs

You can easily script a job which can interact with your configuration management tool for example for requesting automatically the web services and running security scans.

You can see examples in the jobs directory :

`/security-bugracker/security_tools/jobs/run_dependencycheck.php`

`/security-bugracker/security_tools/jobs/run_openvas.php`