

**PANDORA**FMS  
E N T E R P R I S E

**Pandora FMS**  
**Administrator Manual**  
**IMAP Mail Monitoring**



## *Administrator Manual IMAP Mail Monitoring*

© Artica Soluciones Tecnológicas 2005-2013

### **Indice**

1Changelog.....	3
2Introduction.....	4
3Requirements.....	5
4Compatibility Matrix .....	6
5Modules generated.....	7
6Instalation.....	8
7Monitoring.....	10



# 1 CHANGELOG

---

Date	Author	Change	Version
04/07/13	Mario P.	First Version	v1r1

## 2 INTRODUCTION

---

This document describing the email monitoring received from a data filtering.

This plugin perform the following features.

- Monitoring if we get a new mail with filtering that perform (Emitter, Subject, Body text, Date Received...)
- We can capture email information we receive. (numeric data or string)
- You have the option of choosing between different mailbox.
- Support SSL

### 3 REQUIREMENTS

---

The plugin has the requirements to run:

- Have access to our mail server from Pandora server or from other server where you execute the plugin.
- Have Perl installed on your computer.
- Is needed the perl library Mail::Simple and Mail::IMAPClient and for the proper functioning of the plugin.
- In the case of using SSL you should also have the IO::Socket::SSL. These libraries can be installed by default, in the case of not having them can be installed using cpan.

## 4 COMPATIBILITY MATRIX

---

The agent compatibility matrix is the following:

Systems where it has been tested	<ul style="list-style-type: none"> <li>Linux ( SUSE, Ubuntu... )</li> </ul>
Systems where it should work	<ul style="list-style-type: none"> <li>Any system with Perl installed</li> </ul>

## 5 MODULES GENERATED

---

With this plugin you can generate all server modules that you want depending on the filtering setting and all email accounts as needed.

## 6 INSTALATION

Copy the plugin to the Pandora server plugins directory and perform the following performance:

```
perl pandora_imap_v1r1.pl -H host [-p port] -U username -P password -s ( FROM | BODY
| SUBJECT | TEXT | YOUNGER | OLDER ) -s 'string'[-m mailbox][ --capture-data XXXX ]
[-d 0][--ssl --ssl-ca-files XXX --tls ][--imap-retries <tries>] [-v X];
```

To upload it to the server we can do pandora by loading pspz or copying it directly into the server directory Pandora /usr/share/pandora\_server/util/plugin.

The plugin configuration should be this:

The screenshot shows a web-based configuration form for a Pandora plugin. The fields are as follows:

- Name:** Plugin IMAP
- Plug-in command:** /usr/share/pandora\_server/util/plugin/pandora\_imap\_v1r1.pl
- Plug-in type:** Standard (dropdown menu)
- Max. timeout:** (empty text box)
- IP address option:** -H
- Port option:** (empty text box)
- User option:** -U
- Password option:** -P
- Description:** (empty text area)

A 'Create' button with a pencil icon is located at the bottom right of the form.

Once registered the plugin as shown in the image above, or through the pspz to generate a module should follow the guidelines that we indicated in the Monitoring section.

Here you can also find a sample configuration of a module. In this case we will return the number of emails we receive Pandora server for example.

Running the plugin might look like:

```
perl check_imap_receive -H mail.artica.es -U user@artica.es -P pass -s FROM -s
server@pandorafms.com
```



## Module setting in Pandora:

<b>Using module component</b> ?		--Manual setup--	
<b>Name</b>	Chequeo correo PandoraFMS	<b>Disabled</b>	<input type="checkbox"/>
<b>Type</b> ?	Generic numeric	<b>Module group</b>	General
<b>Warning status</b> ?	Min. 0 Max. 0 Str.	<b>Critical status</b> ?	Min. 0 Max. 0 Str.
<b>FF threshold</b> ?	0	<b>Historical data</b>	<input checked="" type="checkbox"/>
<b>Plug-in</b>	Plugin IMAP		
<b>Target IP</b>	mail.artica.es	<b>Port</b>	
<b>Username</b>	user@artica.es	<b>Password</b>	.....
<b>Plug-in parameters</b> ?	-s FROM -s server@pandorafms.com		
<b>Advanced options</b> ↓			
<input type="button" value="Create"/>			

## 7 MONITORING

The plugin aims to monitor emails received through a set of filters that you can set.

Plugin's parameterization:

**-H host:** This shall include the address of the host where the mail server you want to monitor. For example: *mail.artica.es*. Mandatory parameter.

**-p port:** Service port on the IMAP server. Default is 143. If you use SSL, default is 993. If you need setting a different port, you have to indicate in the call of the plugin

**-u user.** We indicate our IMAP mail address. Mandatory parameter. For example: [usuario@artica.es](mailto:usuario@artica.es)

**-p password.** Password of own account. Mandatory parameter. For configure it from the Pandora Console know that this password is encrypted.

**-m mailbox.** The default mailbox is INBOX. In case you want to look elsewhere should be stated. If we dont know the available mailbox, we can run the plugin as follows and will show different mailbox that is in our account:

```
>perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -m -s FROM -s a -v 3
```

Mailbox list:

INBOX

INBOX.Trash

INBOX.old-messages

INBOX.Drafts

INBOX.Sent

**-s ( FROM | BODY | SUBJECT | TEXT | YOUNGER | OLDER | ALL ) -s 'text':** These are all tokens with which we can perform email filter.

- **FROM.** - With this token, we can filter the sender of the email received. Example.

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s FROM -s server@pandorafms.com
```

- **SUBJECT.-** Email Subject Filtering. If you receive such a message from a backup made that has the subject "Backup daily production X" will make this setting in the call.

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s SUBJECT -s "Backup produccion"
```

- **BODY.-** We filter a string of the message body. Example:

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s BODY -s "Text example"
```

We have the ability to link multiple filters such as filter you something of the subject and the body of the message at a time, like this:

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s SUBJECT -s "Backup produccion" -s BODY -s "Text example"
```

- **TEXT.-** Filtering of any text in the mail, and do not know where it will appear.

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s TEXT -s "Text example"
```

- **YOUNGER.-** Time in seconds that can be defined in the call of the plugin to limit the number of messages that will show us the output. This time equates to less than X seconds, for example if we want to show the number of emails received in the last hour we would do this.

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s YOUNGER -s 3600
```

- **OLDER.-** As parameter YOUNGER but in this case when we get the messages received over X time. Over an hour.

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s OLDER -s 3600
```

- **ALL.-** With this token shows all messages.

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s ALL
```

**--capture-data.** By this token you can perform another operation remarkable function of this plugin is to capture the data we receive in the mail. To do this we use regular expressions in its filtering. If for example we received an email with the information from the last backup that was made on a machine, and indicates the disk size occupied what we could get in the following way:

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -s SUBJECT -s "Backup produccion" --capture-data "Backup size: (\d+)"
```

So in this case would have to make generic\_data module type.

**-d 0:** By default the plugin is configured so that each time scan in our email of the emails we have, check that has already shown us and the next execution will not take them into account and we return 0, if who has not received a new mail that matches the filter set. So by this token can configure mails are not marked and so that in each execution all emails show us there in our inbox that meet the filter set.

Example:

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass -d 0 -s FROM -s user2@artica.es
```

**--imap-retries X:** Number of retries to be made in the event that an error occurs in contact with the mail server or the scan when the result is 0. The default value is 1. in the case of adding a high number, it may be that in the case of errors, the execution time increases significantly.

**--ssl:** Enable SSL. Requires IO::Socket::SSL. Using this option automatically changes the default port 143-993. You can still replace the command line with the - port option.

Use with Gmail:

```
perl check_imap_receive -H imap.gmail.com --ssl -U user@gmail.com -P pass -s FROM -s user2@artica.es
```

**--ssl-ca-file XXXX:** Use this option to verify the SSL certificate to a local file server pem. Need to specify the path to the file .pem as a parameter.

Example:

```
perl check_imap_receive -H mail.XX.com -U user@artica.es -P pass --ssl --imap_ssl_cert cert.pem -d 0 -s FROM -s user2@artica.es
```

**--tls:** Enable TLS protocol.

Use with Gmail:

```
perl check_imap_receive -H imap.gmail.com --tls -U user@gmail.com -P pass -s FROM -s user2@artica.es
```