TigaseDoc

Release 0.1

Tigase, Inc.

Jul 02, 2022

CONTENTS

Welcome to Tigase SPAM Filter guide.

CHAPTER

ONE

OVERVIEW

This Tigase SPAM Filter project contains additional features provided for Tigase XMPP Server to reduce number of sent/received SPAM messages.

CHAPTER

TWO

CONFIGURATION

To enable default set of SPAM filters with default settings you need to enable SessionManager processor spam-filter:

Enabling default SPAM filters.

```
'sess-man () {
    'spam-filter' () {}
}
```

2.1 Changing active SPAM filters

You can configure active SPAM filters by setting enabling and disabling SPAM filters (subbeans of spam-filter processor bean).

Enabling message-same-long-body filter.

```
'sess-man' () {
    'spam-filter' () {
        'message-same-long-body' () {}
    }
}
```

2.2 Sending error when packet is dropped

By default, due to nature of SPAM, you do not want to send error packet when SPAM packet is dropped as sending error back will:

- increase traffic on a server (which in rare cases may lead to overload of a XMPP server)
- notify spammer that it was not possible to delivery message

It is possible to configure spam-filter to send error back, by setting true to spam-filter return-error property:

Allow sending error.

```
'sess-man () {
    'spam-filter' () {
        return-error = true
    }
}
```

2.3 Enabling logging of dropped messages

It is possible to enable logging of dropped messages by adding `spam` to comma separate list of values for `--debug` property.

```
--debug=spam
```

CHAPTER

THREE

FILTERS

In this section there is a list of available filters and detailed description of each filtering algorithm.

3.1 Same long message body

When there is a SPAM being sent using XMPP server in most cases number of messages with longer `body` size increases and in most cases every SPAM message contains same `body` part. This filter is identified by following id `message-same-long-body`.

Detection is based on:

- message body being longer that particular value
- multiple messages being sent with same long body

Below is list of possible settings which may be modified to adjust this filter behaviour.

3.1.1 Message body length

SPAM messages are usually longer messages (over 100 chars). To reduce overhead of filtering and memory required for filtering we check length of message body and process it further only if message exceeds declared message body length *(default: 100 chars)*.

You can also check messages with smaller body (ie. only 50 chars) by setting `body-size` property to `50`.

Setting filter to check message with body bigger that 50 chars.

```
'sess-man' () {
    'spam-filter' () {
        'message-same-long-body' () {
            'body-size' = 50
        }
    }
}
```

3.1.2 Number of allowed message with same body

In most cases message with same body is sent to multiple users. Filter will count messages with same body (which is bigger that declared message body length) and if it exceeds message number limit then any further message with same body will be detected and marked as SPAM. By default we allow 20 messages with same body to be processed by SessionManager. If you wish to change this limit set `number-limit` to appropriate value.

Setting number of allowed message to 10.

```
'sess-man' () {
    'spam-filter' () {
        'message-same-long-body' () {
            'number-limit' = 10
        }
    }
}
```

3.1.3 Size of counters cache

We process every message and for every body of message which body length exceeds body length limit we needs to keep counter. This counters are kept in cache which size is configurable and by default equals `10000`. To change size of counters cache assign proper value to `counter-size-limit`.

Increasing cache size to 1000000.

```
'sess-man' () {
    'spam-filter' () {
        'message-same-long-body' () {
            'counter-size-limit' = 1000000
        }
    }
}
```

3.2 Error message and missing <error/> child

Some of SPAM messages are sent as stanzas which are invalid if we compare them with XMPP specification, ie. `<message/>` stanza with `type` attribute set to `error` are sent without child element `<error/>` which is required for all packets of type `error`. This filter detects this kind of messages and marks them as SPAM.

This filter is identified by following id `message-error-ensure-error-child`.

3.3 Groupchat messages sent to bare JID

In some cases SPAM messages are being sent as groupchat messages (messages with `type` attribute set to `groupchat`). With this type of messages we cannot use filtering based on number of message sent with same body as in case of MUC messages we must accept a lot of messages with same body, because there may be many users which are participants of same MUC room and should receive same message.

To address this issue we decided to drop all groupchat messages which are sent to our server XMPP users with `to` attribute set to bare jid, as real MUC component is aware of user resources which joined particular room and will send

messages only to this particular resource by addressing message with full jid. This filter is identified by following id `muc-message-ensure-to-full-jid`.

3.4 Known spammers

To deal with spam it is required to filter every messages to verify if it is spam or not. Usually spammers are using same accounts to send bigger number of messages. This filter takes it as an advantage of this to reduce time required for filtering spam messages as when any other filter marks message as spam this filter will be notified and will mark senders jid as a spammer. This will result in a ban for any packet exchange with this user for configured *ban time*.

If user will send a burst of spam messages then he will be banned for configured ban time for every spam message, ie. if user would send 20 messages and ban time will be set to 15 minutes then users will be banned for 300 minutes (5 hours).

This filter is identified by following id `known-spammers`.

Ban time

Time in minutes for which user marked as spammer will not be able to exchange packets with any other users. By default this value is set to 15 minutes and if you would like to increase it to 30 minutes just add following line to `etc/init.properties` file:

```
'sess-man' () {
    'spam-filter' () {
        'known-spammers' () {
            ban-time = 30
        }
    }
}
```

3.4.1 Cache time

Time in minutes for which user will be remembered as a spammer. It will be able to exchange messages with other users (after ban time passes), but if the situation repeats within this time and our algorithm will be sure that user is a spammer - it may disable local user account.

```
'sess-man' () {
    'spam-filter' () {
        'known-spammers' () {
            cache-time = 10080
        }
    }
}
```

3.4.2 Disabling account

If filter, depending on other filter reports, will establish that user is for sure a spammer it may not only ban user for some time, but it may disable that user account. This is done by default, if you wish to disable account deactivation add following line to `etc/init.properties` file:

```
'sess-man' () {
    'spam-filter' () {
        'known-spammers' () {
            disable-account = false
        }
    }
}
```

3.4.3 Print list of detected spammers

It is possible to request filter to print full list of known spammer which are currently banned every minute. To do so, you need to set `print-spammers` property to `true`.

```
'sess-man' () {
    'spam-filter' () {
        'known-spammers' () {
            print-spammers = true
        }
    }
}
```

3.4.4 Frequency of printing list of spammers

By default, list of detected spammers is printed to logs every day. If you wish you can adjust this value to 1 hour, then add following entry to `etc/init.properties` file:

```
'sess-man' () {
    'spam-filter' () {
        'known-spammers' () {
            print-spammers-frequency = 60
        }
    }
}
```

3.5 Presence subscription filter

When there is a presence-based SPAM being sent using XMPP server in most cases there is a lot of `presence` of type `subscribe` being sent from the single JID. This behavior is annoying and has negative impact on the XMPP server as according to the XMPP specification each `presence` of type `subscribe` sent from JID which is not in the users roster causes adding this JID to the user's roster until user declines subscription request.

Detection is based on counting subscription request being sent from the same bare JID within a period of time.

Below is list of possible settings which may be modified to adjust this filter behaviour.

3.5.1 Number of allowed subscription requests per minute

Be default filter allows 5 subscription requests to be sent from the single JID per minute. If some client will send more than 5 subscription requests it will be marked as a spammer.

Setting filter to allow 7 subscription requests per minute.

```
'sess-man' () {
    'spam-filter' () {
        'presence-subcribe' () {
            'limit-per-minute' = 7
        }
    }
}
```

CHAPTER

FOUR

DEVELOPMENT

You can easily add a new methods of detection if a packet is a spam or not. Simplest way is to implement a new filter.

4.1 Implementation of a new filter

Each class used as a filter by `SpamProcessor` needs to implement `SpamFilter` interface.

There are 3 important methods which need to be implemented by in `SpamFilter` interface:

- `String getId()` - returns id of a filter
- `double getSpamProbability()` - returns probability of sender being a spammer after detection of a single message which is blocked *(from 0.0 to 1.0 where 1.0 means that it is a spammer)*
- `boolean filter(Packet packet, XMPPResourceConnection session)` - method checking if a stanza is a spam (return `false` to stop stanza from being delivered)

Simple filter with id `dummy-detector` which would look for messages with text `dummy`, block them and then mark sender as a spammer after 5 messages would look like this:

Example filter.

```
package test;
import tigase.spam.SpamFilter;

@Bean(name = "dummy-detector", parent = SpamProcessor.class, active = true)
class DummyDetector implements SpamFilter {

    @Override
    public String getId() {
        return "dummy-detector";
    }

    @Override
    public double getSpamProbability() {
        return 0.2;
    }

    @Override
    protected boolean filterPacket(Packet packet, XMPPResourceConnection session) {
        if (packet.getElemName() == "message") {
            Element bodyEl = packet.getElement().getChild("body");
            if (bodyEl != null) {
```

(continues on next page)

(continued from previous page)

```
                String body = bodyEl.getCData();
                if (body != null) {
                    return !body.contains("dummy");
                }
            }
        }
        return true;
    }
}
```

Note: If you expect packet to be processed multiple times (ie. by filter of a sender and filer of a received), then you should take that into account when you estimate value returned by `getSpamProbability()`.

Tip: We have added `@Bean` annotation to automatically enable this filter in the `SpamProcessor` in the Tigase XMPP Server and to be able to easily configure it without specifying full name of a class.