# TigaseDoc

**发行版本 *0.1***

**Tigase, Inc.**

**2022 年 08 月 25 日**

# Contents

欢迎使用 Tigase 垃圾邮件过滤器指南。

CHAPTER 1

# 概述

这个 Tigase 垃圾邮件过滤器项目包含为 Tigase XMPP 服务器提供的附加功能，以减少发送/接收的垃圾邮件消息的数量。

CHAPTER 2

# 配置

要使用默认设置启用默认垃圾邮件过滤器集，您需要启用 SessionManager 处理器垃圾邮件过滤器：

**启用默认垃圾邮件过滤器。**

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

## 2.1 更改活动垃圾邮件过滤器

您可以通过设置启用和禁用垃圾邮件过滤器（垃圾邮件过滤器处理器 bean 的子 bean）来配置活动的垃圾邮件过滤器。

**启用 message-same-long-body 过滤器。**

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

## 2.2 丢包时发送错误

默认情况下，由于垃圾邮件的性质，您不希望在丢弃垃圾邮件数据包时发送错误数据包，因为发送错误返回会：

- 增加服务器上的流量（在极少数情况下可能会导致 XMPP 服务器过载）
- 通知垃圾邮件发送者无法发送邮件

可以通过设置 spam-filter return-error 属性为 true 来配置 spam-filter 以发送错误：

**允许发送错误**。

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

## 2.3 启用丢弃消息的日志记录

可以通过将 `spam` 添加到 `--debug` 属性的逗号分隔值列表中来启用丢弃消息的日志记录。

```
--debug=spam
```

CHAPTER 3

# 过滤器

在本节中，有可用过滤器列表和每种过滤算法的详细说明。

## 3.1 相同的长消息正文

当使用 XMPP 服务器发送垃圾邮件时，在大多数情况下，具有较长 `body` 容量的消息数量会增加，并且在大多数情况下，每条垃圾邮件消息都包含相同的 `body` 部分。此过滤器由以下 id `message-same-long-body` 标识。

检测基于：

- 消息正文比特定值更长
- 使用相同的长正文发送多条消息

下面是可能的设置列表，可以修改这些设置以调整此过滤器行为。

### 3.1.1 消息正文长度

垃圾邮件消息通常是较长的消息（超过 100 个字符）。为了减少过滤所需的过滤和内存开销，我们检查消息正文的长度，并仅在消息超过声明的消息正文长度（默认值：*100* 个字符）时对其进行进一步处理。

您还可以通过将 `body-size` 属性设置为 `50` 来检查具有较小正文（即只有 50 个字符）的消息。

**设置过滤器以检查正文大于 50 个字符的消息。**

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

### 3.1.2 所允许的具有相同正文的消息数

在大多数情况下，具有相同正文的消息会发送给多个用户。过滤器将计算具有相同正文的消息（其大于声明的消息正文长度），如果超过消息数量限制，则将检测到具有相同正文的任何其他消息并将其标记为垃圾邮件。默认情况下，我们允许 SessionManager 处理 20 条具有相同正文的消息。如果您希望更改此限制，请将 `number-limit` 设置为适当的值。

**将允许的消息数量设置为 10**。

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

### 3.1.3 计数器缓存容量

我们处理每条消息，对于每条正文长度超过正文长度限制的消息，我们需要保持计数器。这个计数器保存在缓存中，其大小是可配置的，默认情况下等于 `10000`。要更改计数器缓存的大小，请为 `counter-size-limit` 分配适当的值。

**将缓存大小增加到 1000000**。

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

## 3.2 错误消息和缺少 <error/> child

如果我们将它们与 XMPP 规范进行比较，一些垃圾邮件消息作为无效节发送，即 type 属性设置为 error 的 <message/> 节在没有子元素 <error/> 的情况下发送，这是所有 error 类型的数据包所必需的。此过滤器检测到此类邮件并将其标记为垃圾邮件。

此过滤器由以下 id message-error-ensure-error-child 标识。

## 3.3 群聊消息发送到裸 JID

在某些情况下，垃圾邮件消息作为群聊消息（type 属性设置为 groupchat 的消息）发送。对于这种类型的消息，我们不能使用基于以相同正文发送的消息数量的过滤，因为在 MUC 消息的情况下，我们必须接受大量具有相同正文的消息，因为可能有许多用户是同一个 MUC 房间的参与者，应该收到相同的消息。

为了解决这个问题，我们决定删除所有发送到我们服务器 XMPP 用户的群聊消息，其中 to 属性设置为裸 jid，因为真正的 MUC 组件知道加入特定房间的用户资源，并且只会发送消息到通过使用完整 jid 寻址消息来获取此特定资源。此过滤器由 id muc-message-ensure-to-full-jid 标识。

## 3.4 已知的垃圾邮件发送者

为了处理垃圾邮件，需要过滤每封邮件以验证它是否是垃圾邮件。通常垃圾邮件发送者使用相同的帐户发送更多的消息。此过滤器利用它来减少过滤垃圾邮件所需的时间，因为当任何其他过滤器将邮件标记为垃圾邮件时，此过滤器将被通知并将发件人 jid 标记为垃圾邮件发送者。这将导致在配置的 ban time 内禁止与该用户进行任何数据包交换。

如果用户将发送大量垃圾邮件，他将被禁止为每条垃圾邮件配置禁止时间，即如果用户将发送 20 条消息并将禁止时间设置为 15 分钟，则用户将被禁止 300 分钟（5 小时）。

此过滤器由以下 id known-spammers 标识。

**禁止时间**

用户标记为垃圾邮件发送者的时间（以分钟为单位）里将无法与任何其他用户交换数据包。默认情况下，此值设置为 15 分钟，如果您想将其增加到 30 分钟，只需将以下行添加到 etc/init.properties 文件：

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

### 3.4.1 缓存时间

用户将被记住为垃圾邮件发送者的时间（以分钟为单位）。用户将能够与其他用户交换消息（在禁止时间过去后），但如果在这段时间内情况重复，我们的算法将确定用户是垃圾邮件发送者 - 它可能会禁用本地用户帐户。

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

### 3.4.2 禁用帐户

根据其他过滤器报告，如果过滤器确定用户肯定是垃圾邮件发送者，它可能不仅会禁止用户一段时间，而且可能会禁用该用户帐户。这是默认完成的，如果您希望禁用帐户停用，请在 `etc/init.properties` 文件中添加以下行：

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

### 3.4.3 打印检测到的垃圾邮件发送者列表

可以请求过滤器每分钟打印当前被禁止的已知垃圾邮件发送者的完整列表。为此，您需要将 `print-spammers` 属性设置为 `true`。

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

### 3.4.4 打印垃圾邮件列表的频率

默认情况下，检测到的垃圾邮件发送者列表每天都会打印到日志中。如果您希望可以将此值调整为 1 小时，则将以下条目添加到 `etc/init.properties` 文件中：

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

## 3.5 状态订阅过滤器

在大多数情况下，当使用 XMPP 服务器发送基于存在的垃圾邮件时，会从单个 JID 发送大量 `subscribe` 类型的 `presence`。这种行为很烦人，并且对 XMPP 服务器有负面影响，因为根据 XMPP 规范，从不在用户名册中的 JID 发送的每个 `subscribe` 类型的 `presence` 都会导致将此 JID 添加到用户名册中，直到用户拒绝订阅请求。

检测是基于在一段时间内从同一个裸 JID 发送的订阅请求计数。

下面是可能的设置列表，可以修改这些设置以调整此过滤器行为。

### 3.5.1 每分钟允许的订阅请求数

默认过滤器允许每分钟从单个 JID 发送 5 个订阅请求。如果某些客户端将发送超过 5 个订阅请求，它将被标记为垃圾邮件发送者。

**设置过滤器以允许每分钟 7 个订阅请求**。

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

CHAPTER 4

# 发展

如果数据包是垃圾邮件，您可以轻松添加新的检测方法。最简单的方法是实现一个新的过滤器。

## 4.1 新过滤器的实现

`SpamProcessor` 用作过滤器的每个类都需要实现 `SpamFilter` 接口。

`SpamFilter` 接口中有 3 个重要的方法需要实现：

- `String getId()` - 返回过滤器的 id
- `double getSpamProbability()` - 在检测到一条被阻止的消息后返回发件人成为垃圾邮件发送者的概率 （从 *0.0* 到 *1.0*，其中 *1.0* 表示它是垃圾邮件发送者）
- `boolean filter(Packet packet, XMPPResourceConnection session)` - 检查节是否为垃圾邮件的方法（返回 `false` 以停止传递节）

带有 id `dummy-detector` 的简单过滤器将查找带有文本 `dummy` 的消息，阻止它们，然后在 5 条消息后将发件人标记为垃圾邮件发送者，如下所示：

**示例过滤器**。

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

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

(续下页)

(接上页)

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

**备注:** 如果您希望数据包被处理多次（即通过过滤器过滤发送者和过滤器接收），那么您应该在估计 `getSpamProbability()` 返回的值时考虑到这一点。

**小技巧:** 我们在 Tigase XMPP 服务器的 `SpamProcessor` 中添加了 `@Bean` 注解来自动启用这个过滤器，并且能够在不指定类的全名的情况下轻松配置它。