



# XMPP in real life: attacks, bad behaviour and how to cope with them

2009, 7th february - FOSDEM 2009

Mickaël Rémond <[mremond@process-one.net](mailto:mremond@process-one.net)>

## Background: What we do

 Company created in 1999

## Background: What we do

- 🗨️ Company created in 1999
- 🗨️ 20 employees

## Background: What we do

- 🗨️ Company created in 1999
- 🗨️ 20 employees
- 🗨️ Specialized in Instant Messaging since 2002

## Background: What we do

- 🗨️ Company created in 1999
- 🗨️ 20 employees
- 🗨️ Specialized in Instant Messaging since 2002
- 🗨️ Involved in ejabberd since 2002. ProcessOne produced 98% of the code.

## Background: What we do

- 🗨️ Company created in 1999
- 🗨️ 20 employees
- 🗨️ Specialized in Instant Messaging since 2002
- 🗨️ Involved in ejabberd since 2002. ProcessOne produced 98% of the code.
- 🗨️ Complete stack of Instant Messaging software

## Background: What we do

- 🗨️ Company created in 1999
- 🗨️ 20 employees
- 🗨️ Specialized in Instant Messaging since 2002
- 🗨️ Involved in ejabberd since 2002. ProcessOne produced 98% of the code.
- 🗨️ Complete stack of Instant Messaging software
- 🗨️ Two main activities
  - 🗨️ software: complete software solution for IM
  - 🗨️ expertise: renowned company for high-availability, scalability and custom solutions (consulting, development and hosting)



## Background: What we do

- 🗨️ Company created in 1999
- 🗨️ 20 employees
- 🗨️ Specialized in Instant Messaging since 2002
- 🗨️ Involved in ejabberd since 2002. ProcessOne produced 98% of the code.
- 🗨️ Complete stack of Instant Messaging software
- 🗨️ Two main activities
  - 🗨️ software: complete software solution for IM
  - 🗨️ expertise: renowned company for high-availability, scalability and custom solutions (consulting, development and hosting)
- 🗨️ Several tens of large customers, spread across the world
  - 🗨️ Large scale                      worldwide leader
  - 🗨️ Specific needs                      renowned expertise



# XMPP deployments types

## ☞ Visible public servers

### ☞ The «Sandbox»

☞ Not necessarily large scale but very **unusual** behaviours, clients, usage pattern

# XMPP deployments types

## ☞ Visible public servers

### ☞ The «Sandbox»

☞ Not necessarily large scale but very **unusual behaviours**, clients, usage pattern

## ☞ Large scale servers

☞ Large scale in term of registered or simultaneous users

☞ Large scale starts after a **million of registered** users and / or **hundred of thousands simultaneous** connections

# XMPP deployments types

## ☞ Visible public servers

### ☞ The «Sandbox»

☞ Not necessarily large scale but very **unusual behaviours**, clients, usage pattern

## ☞ Large scale servers

☞ Large scale in term of registered or simultaneous users

☞ Large scale starts after a **million of registered** users and / or **hundred of thousands simultaneous** connections

☞ Large scale in term of throughput

☞ At least tens of thousands of packets per seconds, millions of users of MUC / Pubsub, millions of nodes.

# XMPP deployments types

## ☞ Visible public servers

### ☞ The «Sandbox»

☞ Not necessarily large scale but very **unusual behaviours**, clients, usage pattern

## ☞ Large scale servers

☞ Large scale in term of registered or simultaneous users

☞ Large scale starts after a **million of registered** users and / or **hundred of thousands simultaneous** connections

☞ Large scale in term of throughput

☞ At least **tens of thousands of packets** per seconds or **tens of thousands users** in MUC room or subscribed to pubsub node, etc.

☞ Experience of **large clusters** with several **tens of millions registered users** and more than **500 000 simultaneous users**.

# Challenges of real life XMPP

 Uptime

# Challenges of real life XMPP

 Uptime

 Uptime !

# Challenges of real life XMPP

🗨️ Uptime

🗨️ Uptime !

🗨️ Uptime !!



# Challenges of real life XMPP

☞ Uptime

☞ Uptime !

☞ Uptime !!

**Everything else derives from this  
Challenge (performance, scalability)**

# Challenges of real life XMPP

☞ Uptime

☞ Uptime !

☞ Uptime !!

Everything else derives from this  
Challenge (performance, scalability)

☞ When a server is restarted:

- ☞ it faces a reconnect storm from client that login again

- ☞ it needs to resync the complete presence states with most of its known s2s servers

- ☞ it reconnects the users accounts on gateways ...

# Challenges of real life XMPP

☞ Uptime

☞ Uptime !

☞ Uptime !!

Everything else derives from this  
Challenge (performance, scalability)

☞ When a server is restarted:

☞ it faces a reconnect storm from client that login again

☞ it needs to resync the complete presence states with most of its known s2s servers

☞ it reconnects the users accounts on gateways ...

☞ You need to:

☞ Be able to **monitor** lots of values to **detect troubles** and have tools to keep the server online during trouble phase (otherwise it crash: get worse)

☞ Be able to perform **maintenance** task and **upgrade code** live

## Case 1: XMPP as a proxy

🗨️ Symptom: A «*sandbox*» XMPP server crashes regularly

## Case 1: XMPP as a proxy

- 🗨 Symptom: A «*sandbox*» XMPP server crashes regularly
- 🗨 First challenge: Detect possible abuser

## Case 1: XMPP as a proxy

- 🗨️ Symptom: A «sandbox» XMPP server crashes regularly
- 🗨️ First challenge: Detect possible abuser
- 🗨️ Use of our toolkit (TeamLeader console) to analyse traffic patterns.

## Case 1: XMPP as a proxy

- ☞ Symptom: A «sandbox» XMPP server crashes regularly
- ☞ First challenge: Detect possible abuser
- ☞ Use of our toolkit (TeamLeader console) to analyse traffic patterns.
- ☞ Correlate the crash to a given user
  - ☞ Large number of packets send when online
  - ☞ Large bandwidth consumption



## Case 1: XMPP as a proxy

- ☞ Symptom: A «sandbox» XMPP server crashes regularly
- ☞ First challenge: Detect possible abuser
- ☞ Use of our toolkit (TeamLeader console) to analyse traffic patterns.
- ☞ Correlate the crash to a given user
  - ☞ Large number of packets send when online
  - ☞ Large bandwidth consumption
- ☞ Dump traffic of this user for analysis

## Case 1: XMPP as a proxy

- ☞ Symptom: A «sandbox» XMPP server crashes regularly
- ☞ First challenge: Detect possible abuser
- ☞ Use of our toolkit (TeamLeader console) to analyse traffic patterns.
- ☞ Correlate the crash to a given user
  - ☞ Large number of packets send when online
  - ☞ Large bandwidth consumption
- ☞ Dump traffic of this user for analysis
- ☞ Traffic reveals that user:
  - ☞ has deployed XMPP bot at work on his servers
  - ☞ is using the public server to get control of his server
  - ☞ basically «Shell over XMPP»

## Case 1: XMPP as a proxy

- ☞ Symptom: A «sandbox» XMPP server crashes regularly
- ☞ First challenge: Detect possible abuser
- ☞ Use of our toolkit (TeamLeader console) to analyse traffic patterns.
- ☞ Correlate the crash to a given user
  - ☞ Large number of packets send when online
  - ☞ Large bandwidth consumption
- ☞ Dump traffic of this user for analysis
- ☞ Traffic reveals that user:
  - ☞ has deployed XMPP bot at work on his servers
  - ☞ is using the public server to get control of his server
  - ☞ basically «Shell over XMPP»
- ☞ Response: Need to detect abnormal usage pattern and trigger alerts

## Case 2: Client bad behaviours

🗨 Symptom: Abnormal memory consumption / sometime leading to crash

## Case 2: Client bad behaviours

- 🗨️ Symptom: Abnormal memory consumption / sometime leading to crash
- 🗨️ Source problem had been client behaviour

## Case 2: Client bad behaviours

- ☞ Symptom: Abnormal memory consumption / sometime leading to crash
- ☞ Source problem had been client behaviour
- ☞ Generates an undue load on the server
  - ☞ Example: Client does not reply to some IQ stanzas (PEPS / CAPS)
    - ☞ Server waits for reply until timeout
    - ☞ Depending on the type of processing it can be blocking
  - ☞ Example: Client that send too many presences
    - ☞ Large presence broadcast, especially in MUC rooms



## Case 2: Client bad behaviours

- ☞ Symptom: Abnormal memory consumption / sometime leading to crash
- ☞ Source problem had been client behaviour
- ☞ Generates an undue load on the server
  - ☞ Example: Client does not reply to some IQ stanzas (PEPS / CAPS)
    - ☞ Server waits for reply until timeout
    - ☞ Depending on the type of processing it can be blocking
  - ☞ Example: Client that send too many presences
    - ☞ Large presence broadcast, especially in MUC rooms
- ☞ Need to restrict the ability to perform those patterns:
  - ☞ Limit the interval for sending presences in chat rooms
  - ☞ Limit resourc consumption in general



## Case 2: Client bad behaviours

- ☞ Symptom: Abnormal memory consumption / sometime leading to crash
- ☞ Source problem had been client behaviour
- ☞ Generates an undue load on the server
  - ☞ Example: Client does not reply to some IQ stanzas (PEPS / CAPS)
    - ☞ Server waits for reply until timeout
    - ☞ Depending on the type of processing it can be blocking
  - ☞ Example: Client that send too many presences
    - ☞ Large presence broadcast, especially in MUC rooms
- ☞ Need to restrict the ability to perform those patterns:
  - ☞ Limit the interval for sending presences in chat rooms
  - ☞ Limit resourc consumption in general

## Case 3: Multi User chat

- MUC rooms attacks

- Most common case of abuse

## Case 3: Multi User chat

### 🗨️ MUC rooms attacks

- 🗨️ Most common case of abuse

- 🗨️ Create a lot of MUC persistent MUC rooms





## Case 3: Multi User chat

### 🗨️ MUC rooms attacks

- 🗨️ Most common case of abuse
- 🗨️ Create a lot of MUC persistent MUC rooms
- 🗨️ Join a lot of MUC rooms

## Case 3: Multi User chat

### MUC rooms attacks

-  Most common case of abuse
-  Create a lot of MUC persistent MUC rooms
-  Join a lot of MUC rooms
-  Join / leave a MUC room fastly

## Case 3: Multi User chat

### 🗨️ MUC rooms attacks

- 🗨️ Most common case of abuse
- 🗨️ Create a lot of MUC persistent MUC rooms
- 🗨️ Join a lot of MUC rooms
- 🗨️ Join / leave a MUC room fastly
- 🗨️ Join lots of users in a single room

## Case 3: Multi User chat

### 🗨️ MUC rooms attacks

- 🗨️ Most common case of abuse
- 🗨️ Create a lot of MUC persistent MUC rooms
- 🗨️ Join a lot of MUC rooms
- 🗨️ Join / leave a MUC room fastly
- 🗨️ Join lots of users in a single room
- 🗨️ Change presence to bypass voice



## Case 3: Multi User chat

### 🗨️ MUC rooms attacks

- 🗨️ Most common case of abuse
- 🗨️ Create a lot of MUC persistent MUC rooms
- 🗨️ Join a lot of MUC rooms
- 🗨️ Join / leave a MUC room fastly
- 🗨️ Join lots of users in a single room
- 🗨️ Change presence to bypass voice
- 🗨️ Flood with messages

## Case 3: Multi User chat

### 🗨️ MUC rooms attacks

- 🗨️ Most common case of abuse
- 🗨️ Create a lot of MUC persistent MUC rooms
- 🗨️ Join a lot of MUC rooms
- 🗨️ Join / leave a MUC room fastly
- 🗨️ Join lots of users in a single room
- 🗨️ Change presence to bypass voice
- 🗨️ Flood with messages
- 🗨️ Use large values to «attack» the server or the client (large room names, large nick names, etc)

## Case 4: Bots

 Symptom: Server crash

## Case 4: Bots

🗨️ Symptom: Server crash

🗨️ Reduce to a crash when some special user connects (every time)

## Case 4: Bots

- 🗨️ Symptom: Server crash
- 🗨️ Reduce to a crash when some special user connects (every time)
- 🗨️ Bots send messages to their users on a public server

## Case 4: Bots

- 🗨️ Symptom: Server crash
- 🗨️ Reduce to a crash when some special user connects (every time)
- 🗨️ Bots send messages to their users on a public server
- 🗨️ They do not often use headline message type (which means they are not intended to be stored offline).
- 🗨️ They sometimes rely on presence, but it can be inaccurate after a force server shutdown.
- 🗨️ We have seen users of public servers with more than 500 000 messages in the offline store.

## Case 4: Bots

- ☞ Symptom: Server crash
  - ☞ Reduce to a crash when some special user connects (every time)
  - ☞ Bots send messages to their users on a public server
  - ☞ They do not often use headline message type (which means they are not intended to be stored offline).
  - ☞ They sometimes rely on presence, but it can be inaccurate after a force server shutdown.
  - ☞ We have seen users of public servers with more than 500 000 messages in the offline store.
- 
- ☞ Limit the size of the offline store
  - ☞ Ability to detect abusers and limit their ability to send massive amount of messages



## Case 5: Large flow / small pipes

- 🗨️ An XMPP server is a pipe
- 🗨️ Data **flows** from one connection to another.

## Case 5: Large flow / small pipes

- 🗨️ An XMPP server is a pipe
- 🗨️ Data **flows** from one connection to another.
- 🗨️ Problem:
  - 🗨️ What happens if you try to send data faster than the target client can receive (mobile) ?
  - 🗨️ What happens if you try to send data faster than the target server can receive (limited bandwidth, Karma limitation) ?

## Case 5: Large flow / small pipes

🗨️ An XMPP server is a pipe

🗨️ Data **flows** from one connection to another.

🗨️ Problem:

🗨️ What happens if you try to send data faster than the target client can receive (mobile) ?

🗨️ What happens if you try to send data faster than the target server can receive (limited bandwidth, Karma limitation) ?

🗨️ Challenge:

🗨️ Detect congestions and decide what to do when this happens

🗨️ This has to be done right otherwise the service might seem unreliable

🗨️ Federation rules / pattern needed ?

## Interesting new challenges ahead

- Massive numbers of XMPP servers deployed
  - Lots of s2s connections to maintain for large servers
  - Will XMPP scale to millions of servers ?

## Interesting new challenges ahead

- ☞ Massive numbers of XMPP servers deployed
  - ☞ Lots of s2s connections to maintain for large servers
  - ☞ Will XMPP scale to millions of servers ?
- ☞ Large servers connected through s2s:
  - ☞ Several large servers need to keep users presence in sync
  - ☞ Imagine what happen when one of them goes down ...
    - ☞ Yes, massive presence resync is needed

## Interesting new challenges ahead

- ☞ Massive numbers of XMPP servers deployed
  - ☞ Lots of s2s connections to maintain for large servers
  - ☞ Will XMPP scale to millions of servers ?
- ☞ Large servers connected through s2s:
  - ☞ Several large servers need to keep users presence in sync
  - ☞ Imagine what happen when one of them goes down ...
    - ☞ Yes, massive presence resync is needed
- ☞ New usage patterns
  - ☞ Ubiquitous XMPP: A single users can have many connections: Increase in size of XMPP platforms
  - ☞ Devices / Machine to Machine communication: increase of volume of messages





## Questions and challenges to share ?