# Incident Reporting and XMPP

Peter Saint-Andre

SIIS Workshop

2013-07-26, Berlin

# About Me

- Author of the XMPP RFCs and numerous XMPP extensions
- Co-author of "XMPP: The Definitive Guide" (O'Reilly, 2009)
- Operator of jabber.org IM service (1+ million users)
- More about me at https://stpeter.im/

# What is XMPP?

- Application protocol for streaming XML
- Grew out of Jabber open-source community starting in 1999
- Widely used for IM, chat, presence, notifications, m2m, IoT, etc.
- Formalized as Extensible Messaging and Presence Protocol (RFC 3920 / RFC 6120)

# What Does XMPP Provide?

- Messaging (1:1, 1:many, many:many)
- Channel encryption (TLS)
- Strong authentication (SASL) and identity
- Interdomain federation
- Service discovery + device capabilities
- Lots of protocol extensions @ xmpp.org

# Not Just "Rough Consensus"

- 100+ software implementations
- 100,000+ service deployments (both public and private)
- A large public network of XMPP servers (anyone can install server software and connect to the network)

# Incidents on the XMPP Network

- Unfortunately, if you run a large public network you'll experience "incidents"
- In XMPP, most common incidents are:
  - Chatroom flooders
  - Spammy users (especially presence subscription spam)
- Distributed network makes control hard

# Incident Reporting for the XMPP Network

- XMPP servers share reports and request actions (e.g., disable account for flooding)
- Originally (2009) defined a custom format
- Eventually (2012) updated to use IODEF (RFC 5070)
- Not implemented or deployed yet :(

# XEP-0268

- http://xmpp.org/extensions/xep-0268.html
- Uses XMPP <iq/> stanza between servers, whitelist based on trust relationship
- 3 child elements: <report/>, <request/>, <response/>
- Each contains an IODEF <Incident/>
- IODEF extensions for XMPP-specific data

# Generalized Incident Reporting Using XMPP

- Could we use XMPP as a transport for information sharing? Two approaches:
  - Basic: a "firehose" with a single live feed per reporting entity (maybe similar to AbuseHelper project?)
  - Advanced: multiple feeds per reporting entity based on report type, data format, intended audience, etc.

# Basic Approach

- PRO: very simple, all XMPP servers support interdomain federation, whitelist by domain based on trust relationship
- CON: all reports are sent to all parties, no easy way to (a) control what is reported to whom or (b) filter out what you receive

# Advanced Approach

- PRO: fine-grained control over both data sharing and data access, with several authorization models
- CON: depends on XMPP Publish-Subscribe extension (XEP-0060), which is not yet supported in all servers / code libraries

# Next Steps

- Deploy XEP-0268 to enable "self-healing" of the XMPP network
- Determine whether generalized incident reporting over XMPP might solve some new and interesting problems
  - Basic or advanced? Join the conversation to help us figure that out!