Jabber security

Peter Saint-Andre

stpeter@jabber.org


pd

secure communications

with Jabber

Jabber is....

open technologies

real-time messaging

presence

multimedia negotiation

and more

invented by Jeremie Miller in 1998

powered by
streaming XML

over long-lived
TCP connections

client-server architecture

decentralized
network

inter-domain messaging

like email

but really fast

with built-in presence

not one open-source project

multiple codebases

open-source and commercial

focus on XML wire protocol

core protocol
standardized @ IETF

Extensible

Messaging

and

Presence

Protocol

(XMPP)

RFCs 3920 + 3921

widely deployed

how many users?

we don’t know

decentralized architecture

~50 million IM users

not just IM

general XML routing

lots of applications
beyond IM

continually defining
XMPP extensions

XMPP Standards Foundation (XSF)

that’s great, but...

how secure is it?

what is security?

secure conversation in real life...

a good friend
visits your home

you know and trust
each other

only the two of you

strangers can’t enter
your home

your home is not bugged

conversation is not recorded

what you say is private
and confidential

contrast with the Internet...

the Internet is a dangerous place

lots of potential attacks

man in the middle

unauthenticated users

address spoofing

weak identity

rogue servers

denial of service

directory harvesting

buffer overflows

spam

spim

spit

splogs

viruses

worms

trojan horses

malware

phishing

pharming

information leaks

inappropriate logging and archiving

etc.

how do we fight
these threats?

sorry, but...

Jabber is not a perfect technology

not originally built
for high security

don’t require GPG keys
or X.509 certs

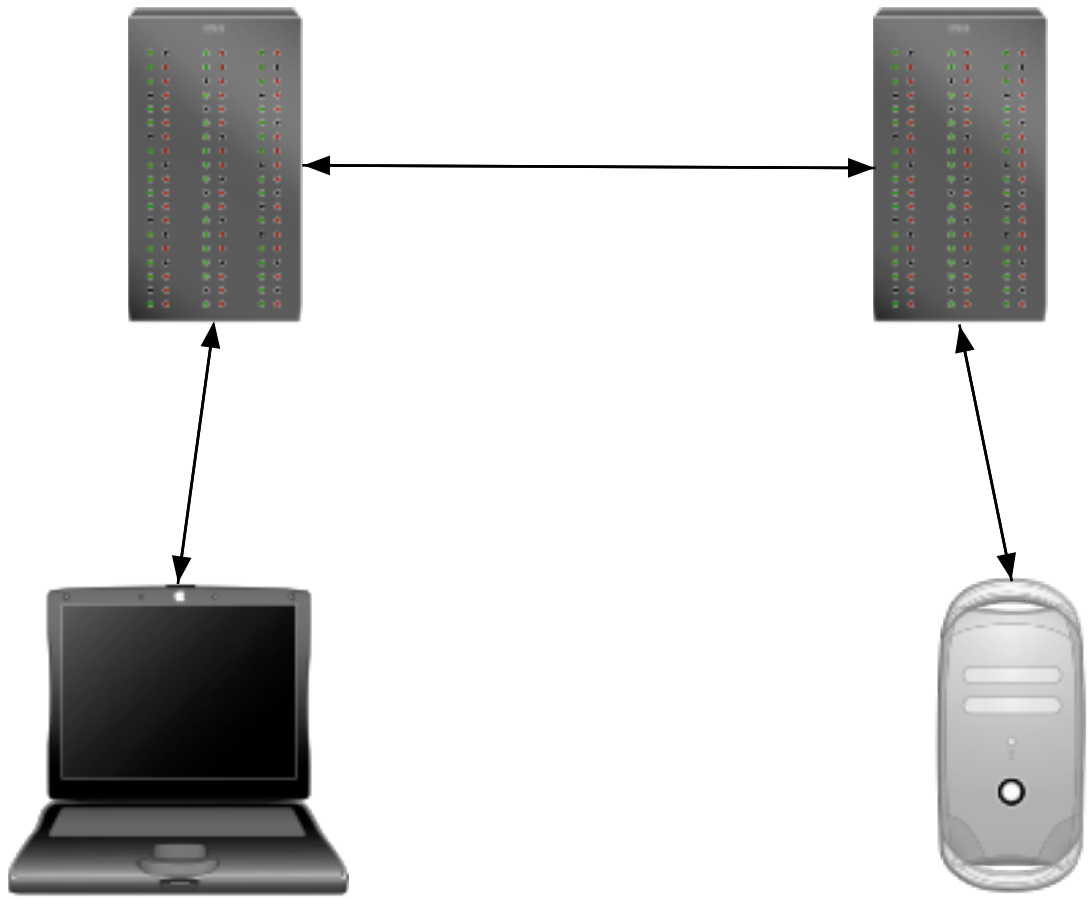
don’t require ubiquitous encryption

maybe that's why we have 50 million users...

but privacy and security are important

so what have we done to help?

Jabber architecture...



client-server architecture

similar to email

client connects to server (TCP 5222)

(or connect via HTTP binding over SSL)

client MUST
authenticate

originally: plaintext or hashed password

Simple Authentication & Security Layer (SASL)

RFC 4422

many SASL mechanisms

PLAIN (OK over encrypted connection)

DIGEST-MD5

EXTERNAL (with X.509 certs)

KERBEROS
(a.k.a. GSSAPI)

ANONYMOUS

etc.

all users are authenticated

server stamps
user ‘from’ address

Jabber IDs are logical addresses

look like email addresses

romeo@montague.net

juliet@capulet.com

not limited to
US-ASCII characters

jiři@čechy.cz

πλατω@έλλας.gr

มฌำปจ@jabber.th

ぷぉぞ@jabber.jp

∞@math.it

full Unicode opens phishing attacks

STPETER@jabber.org
STPETER@jabber.org

clients should use
“petnames”

store in buddy list [tm]
(a.k.a. “roster”)

server stores
your roster

server broadcasts
your presence

but only to subscribers
you have authorized

server must not expose your IP address

most traffic goes through server

traffic is pure XML

servers reject
malformed XML

servers MAY validate traffic against schemas

difficult to inject
binary objects

difficult to propagate malware

break alliance between viruses and spam

spim virtually unknown
on Jabber network

why?

hard to spoof addresses

hard to send
inline binary

XHTML subset
(no scripts etc.)

clients check before accepting a file

XMPP not immune to spim

have spim-fighting tools ready when it appears

challenge-response to communicate

challenge-response to register account

spim reporting

reputation systems?

spimmers need to overcome rate limiting

distributed attack or
rogue server

not impossible

just harder than other networks (got email?)

no rogue servers
(yet)

a server MAY federate with other servers

many private
XMPP servers

public servers federate as needed (TCP 5269)

DNS lookups to determine IP addresses

only one hop
between servers

server identities
are validated

server dialback
(reverse DNS lookups)

effectively prevents
server spoofing

receiving server checks sending domain

no messages from
“service@paypal.com”

DNS poisoning
can invalidate

need something stronger?

Transport Layer Security (TLS)

RFC 4346

IETF “upgrade” to SSL

TLS + SASL EXTERNAL with X.509 certs

strong authentication
of other servers

but only if not using self-signed certs

$$$

real X.509 certs
are expensive

free digital certificates for XMPP server admins

intermediate CA for XMPP network

xmpp.net

root CA: StartCom

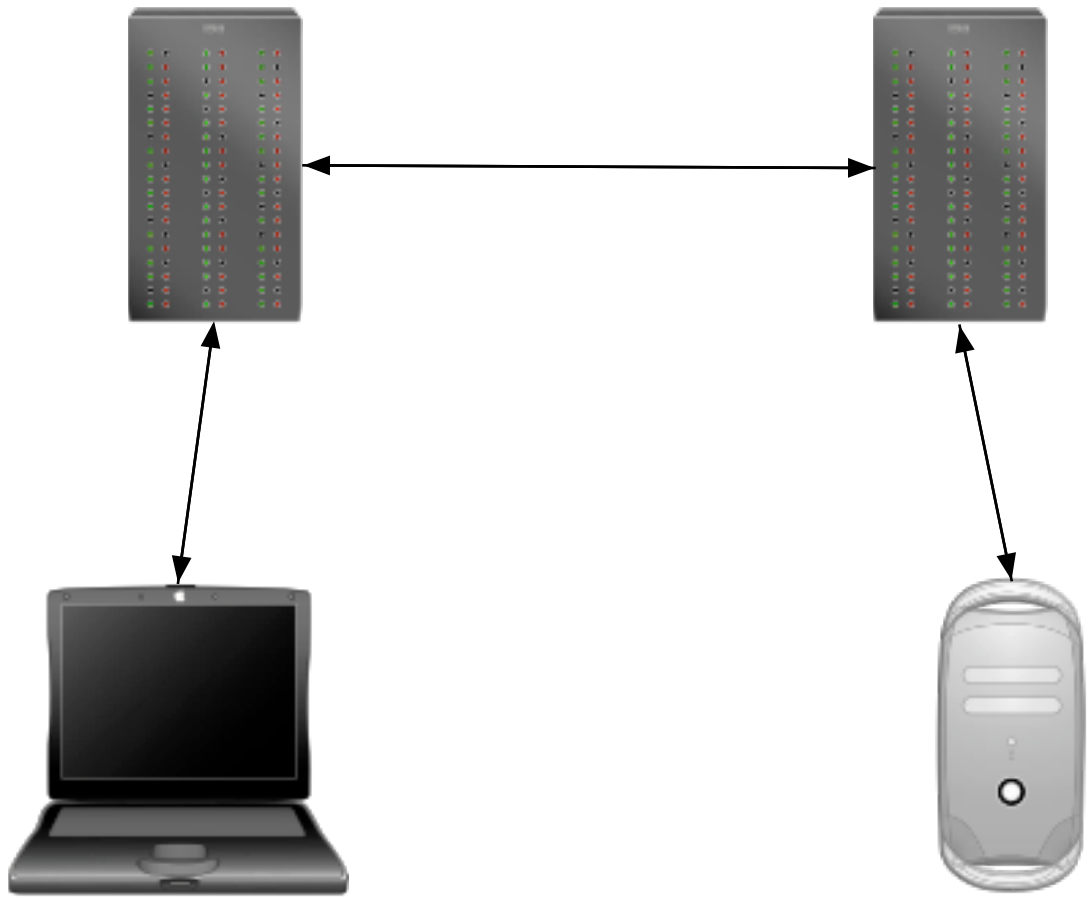
ICA: XMPP Standards Foundation

hopefully other CAs
in future

channel encryption
is a no-brainer

Mallory is foiled

but what about
Isaac and Justin?



need end-to-end encryption (“e2e”)

first try: OpenPGP (XEP-0027)

great for geeks

but Aunt Tillie
doesn't use PGP

second try: S/MIME (RFC 3923)

great for geeks (and some employees)

but Aunt Tillie
doesn't use X.509

XML encryption and digital signatures?

seems natural, but not much interest (c14n?)

doesn’t provide perfect forward secrecy

off-the-record communications (OTR)?

great idea

opportunistic encryption (à la SSH)

perfect forward secrecy

but encrypts only the plaintext message body

we need to encrypt
the entire packet

why?

because XMPP is more than just IM

e.g., protect IPs sent in multimedia negotiation

solution: encrypted sessions

big set of requirements...

packets are confidential

packet integrity

replay protection

key compromise does not reveal past comms

dependence on PKI
not necessary

entities authenticated
to each other

3rd parties cannot identify entities

repudiate any given message

robustness against
attack (multiple hurdles)

upgradeability if bugs are discovered

encryption of full XMPP packets

implementable by
typical developer

usable by
typical user

just a dream?

how to address all requirements?

bootstrap from cleartext to encryption

in-band Diffie-Hellman key exchange

translate SIGMA approach to XMPP

similar to Internet Key Exchange (IKE)

details in XSF XEPs
116, 188, 200

major priority for 2007

support from NLnet
(thanks!)

pursuing full security analysis

code bounties

more at
blog.xmpp.org

wide implementation
by end of 2007

so how are we doing?

spim free

hard to spoof addresses

pure XML discourages binary malware

DoS attacks possible
but not easy

widespread channel encryption

working hard on
end-to-end encryption

widely deployed in high-security environments

Wall Street investment banks

U.S. military

MIT and other universities

many public servers
since 1999

no major security breaches

can’t be complacent

always more to do

security is a never-ending process

analysis and hacking encouraged

if it breaks,
we'll fix it

security@xmpp.org

join the conversation

let's build
a more secure Internet