XEP-0165: Best Practices to Discourage JID Mimicking

Peter Saint-Andre
mailto:peter@andyet.net
xmpp:stpeter@stpeter.im
https://stpeter.im/

2007-12-13
Version 0.6

Status	Type	Short Name
Deferred	Informational	N/A

This document recommends best practices to discourage mimicking of Jabber IDs.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 - 2016 by the XMPP Standards Foundation (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1 Introduction

There are two forms of address spoofing: forging and mimicking.
In the context of Jabber/XMPP technologies, an address is *forged* when an entity is able to generate an XML stanza whose 'from' address does not correspond to the account credentials with which the entity authenticated onto the network -- for example, if an entity that authenticated as "stpeter@jabber.org" is able to send XML stanzas from "MaineBoy@jabber.org" or "peter@saint-andre.com".
Address forging is difficult in Jabber/XMPP systems given the requirement for sending servers to stamp 'from' addresses and for receiving servers to verify sending domains via server dialback or server-to-server authentication (see XMPP Core [1]). Difficult, but not impossible: a rogue server could forge JIDs at the sending domain by ignoring the stamping requirement and could even forge JIDs at other domains by means of a DNS poisoning attack. However, discussion of ways to deal with such rogue servers is out of scope for this document.
An address is *mimicked* when an entity provides legitimate authentication credentials for and sends XML stanzas from an account whose Jabber ID (JID) appears to a human user to be the same as another JID -- for example, in some clients "paypa1@jabber.org" (spelled with the number one as the final character of the node identifier) may appear to be the same as "paypal@jabber.org (spelled with the lower-case version of the letter "L"). [2] A more sophisticated example of address mimicking (which may not render correctly in all browsers) is the following:

```
@.org
```

That JID is not an uppercase version of "stpeter@jabber.org" in US-ASCII characters, but a fake JID made up mostly of Cherokee characters, namely:

```
U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2
  @
  U+13AB U+13AA U+13F4 U+13F4 U+13AC U+13D2 .org
```

In this example, it is unlikely that the average user could tell the difference between the real JID and the fake JID. [3]
By contrast with address forging, it may be relatively easy to mimic (some) JIDs in Jabber/XMPP systems, especially because JIDs can contain almost any Unicode character. The possibility of address mimicking introduces security vulnerabilities of the kind that have also plagued the World Wide Web, specifically the phenomenon known as phishing. [4] To combat

[1] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[2] This phenomenon is sometimes called "typejacking".
[3] Naturally, there is no way to distinguish with full certainty which is the fake JID and which is the real JID. For example, in some communication contexts, the Cherokee JID may be the real JID and the US-ASCII JID may thus appear to be the fake JID.
[4] Phishing has been defined by the Financial Services Technology Consortium Counter-Phishing Initiative as "a broadly launched social engineering attack in which an electronic identity is misrepresented in an attempt

those vulnerabilities, this document recommends a set of best practices to minimize the potential impact of address mimicking on the Jabber/XMPP network. [5]

2 Recommendations

2.1 Presentation of JIDs

Every human user of Jabber/XMPP technologies presumably has a preferred language (or, in some cases, a small set of preferred languages), which an XMPP application SHOULD gather either explicitly from the user or implicitly via the operating system of the user's device. Furthermore, every language has a range of characters normally used to represent that language in textual form. Therefore, an XMPP application SHOULD warn the user when presenting a JID that uses characters outside the normal range of the user's preferred language(s). [6]

2.2 The Roster as a Petname System

As explained in Introduction to Petname Systems [7], no one naming or address scheme can provide names that are simultaneously global, memorable, and unique. However, certain *combinations* of names and addresses can provide all three properties, and such combinations are commonly called "petname systems". In particular, the information contained in a user's roster (see XMPP IM [8]) can be combined with information provided by a user's contacts to construct a petname system. Consider the following combination of names:

1. The JID "stpeter@jabber.org" is globally unique on the Jabber/XMPP network, but it is not necessarily memorable.

2. The nickname "psa" (asserted by the user associated with the address "stpeter@jabber.org") is globally memorable but not necessarily unique; see User Nickname (XEP-0172) [9] for more information about user-asserted nicknames.

to trick individuals into revealing personal credentials that can be used fraudulently against them"). To be precise, the current document (1) does not assume that such attacks will be broadly launched and (2) focuses on the misrepresentation of Jabber IDs (not any other identifiers) within the context of Jabber/XMPP systems.

[5] This document does not cover handling of non-XMPP addresses, for example HTTP URLs. Jabber/XMPP clients SHOULD handle such addresses in accordance with best practices for the relevant non-XMPP technology.

[6] This recommendation is not intended to discourage communication across language communities; instead, it simply recognizes the existence of such language communities and encourages due caution when presenting unfamiliar character sets to human users.

[7] Introduction to Petname Systems <http://www.skyhunter.com/marcs/petnames/IntroPetNames.html>.

[8] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.

[9] XEP-0172: User Nickname <http://xmpp.org/extensions/xep-0172.html>.

3. The handle or petname "that protocol dude" (assigned by a contact who adds "stpeter@jabber.org" to her contact list) is privately memorable and unique [10] but is by no means global since it has meaning only to the person who assigns it; for consistency with XEP-0172 and XMPP IM [11] we refer to this as a "handle". [12]

A client SHOULD require an end user to assign a handle for every contact added to the person's roster, which SHOULD be stored in the roster as the value of the <item/> element's 'name' attribute (see the Security Considerations section of this document for further discussion). A client SHOULD then present that handle instead of or in addition to the contact's JID or nickname (e.g., in the user's roster and in chat interfaces). This will help to discourage mimicked addresses from being presented as equivalent to the address that is being mimicked.

2.3 Associating Security Credentials with Roster Items

Although a Jabber ID can be considered globally unique, the petname system in which it is embedded can be strengthened by associating that JID with a key that can be used for signing and encryption (such as an OpenPGP key, X.509 certificate, or RSA key), preferably a key that encapsulates the associated XMPP address. A client SHOULD associate a key with the user of that client, and SHOULD generate such a key if the user does not have one.
Unfortunately, cryptographic identities such as keys, certificates, and fingerprints are even less memorable than JIDs, which makes assigning a handle even more important. Therefore, if a contact provides such a cryptographic identity, a client MUST reliably associate it with the contact in a user's roster (including, as mentioned, a handle for each contact) in order to further strengthen the petname system.

2.4 Referrals

In order to strengthen the web of interaction and trust between Jabber/XMPP users, it is helpful for them to share roster items. In particular, when a user wants to subscribe to the presence of a potential contact, the user SHOULD seek a referral from a third person who knows both the user and the contact. Such a referral consists of a roster item sent from the third person to the potential contact, encapsulated using the Roster Item Exchange (XEP-0144) [13] protocol:

Listing 1: A Basic Referral

```
<message from='peter@saint-andre.com' to='MaineBoy@jabber.org'>
```

[10] If not shared or leaked, it may even be securely unique.
[11] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.
[12] In RFC 6121 this was referred to as an "alias".
[13] XEP-0144: Roster Item Exchange <http://xmpp.org/extensions/xep-0144.html>.

```
  <x xmlns='http://jabber.org/protocol/rosterx'>
    <item jid='stpeter@jabber.org' name='Peter␣Saint-Andre'/>
  </x>
</message>
```

Here, the 'name' attribute encapsulates what in petname systems is known as an "alleged name", that is, the name for an entity proposed by a third party.
Such a referral SHOULD also include the user's nick as understood by the third person (encapsulated in the format defined in XEP-0172) and fingerprint of the user as understood by the third person (encapsulated in the format defined in Public Key Publishing (XEP-0189) [14]):

Listing 2: Referral With Nickname and Public Key

```
<message from='peter@saint-andre.com' to='MaineBoy@jabber.org'>
  <x xmlns='http://jabber.org/protocol/rosterx'>
    <item jid='stpeter@jabber.org' name='Peter␣Saint-Andre'>
      <nick xmlns='http://jabber.org/protocol/nick'>psa</nick>
      <KeyInfo xmlns='http://www.w3.org/2000/09/xmldsig#'>
        <KeyName>stpeterRSAkey1</KeyName>
        ...
      </KeyInfo>
      <KeyInfo xmlns='http://www.w3.org/2000/09/xmldsig#'>
        <KeyName>stpeterX509cert1</KeyName>
        ...
      </KeyInfo>
    </item>
  </x>
</message>
```

The third person MUST NOT simply copy the key as communicated by the user but instead MUST validate it against the public key of the user.

2.5 Subscription Requests

We have seen that, at a minimum, three names or address types are needed to provide a petname system for XMPP: a JID, a nickname, and a handle (preferably strengthened by inclusion of a fingerprint derived from a key). However, at present a subscription request contains only the JID of the sender:

Listing 3: A Basic Subscription Request

```
<presence from='stpeter@jabber.org' to='MaineBoy@jabber.org' type='
    subscribe'/>
```

[14] XEP-0189: Public Key Publishing <http://xmpp.org/extensions/xep-0189.html>.

Naturally, based on the JID, it is possible to pull information about the sender from a persistent data store such as an LDAP database, vcard-temp (XEP-0054) [15] node, or future profile system (see User Profile (XEP-0154) [16]). However, to speed interactions, this document recommends that when a client sends a subscription request, it SHOULD include the preferred nickname of the sender (encapsulated via the format specified in XEP-0172) and the sender's key or keys.

Listing 4: Subscription Request With Nickname and Key

```
<presence from='stpeter@jabber.org' to='MaineBoy@jabber.org' type='
    subscribe'>
  <nick xmlns='http://jabber.org/protocol/nick'>psa</nick>
  <KeyInfo xmlns='http://www.w3.org/2000/09/xmldsig#'>
    <KeyName>stpeterRSAkey1</KeyName>
    ...
  </KeyInfo>
  <KeyInfo xmlns='http://www.w3.org/2000/09/xmldsig#'>
    <KeyName>stpeterX509cert1</KeyName>
    ...
  </KeyInfo>
</presence>
```

If one or more referrals have been received, the user or client MUST check the key or keys provided in the subscription request against the key or keys provided in the referral or referrals.

3 Security Considerations

A client should not allow a user to assign as a handle the alleged name received in a referral. In order for a user-assigned handle to strengthen the security of the petname system, the user must not shared the handle with other individuals. If the handle is stored in the user's roster, the handle may be compromised since roster storage is not necessarily a secure medium (e.g., the handle could be read by a server administrator). If the server is not trusted by the user, the client should store the handle locally on the user's device rather than in the roster.
A user should not place more trust in a referral than he or she places in the person who sends it.

4 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [17].

[15] XEP-0054: vcard-temp <http://xmpp.org/extensions/xep-0054.html>.
[16] XEP-0154: User Profile <http://xmpp.org/extensions/xep-0154.html>.
[17] The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see

5 XMPP Registrar Considerations

This document requires no interaction with the XMPP Registrar [18].

<http://www.iana.org/>.

[18] The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <http://xmpp.org/registrar/>.