Internet Engineering Task Force (IETF)
Request for Comments: 7622
Obsoletes: 6122
Category: Standards Track
ISSN: 2070-1721

P. Saint-Andre
&yet

September 2015

Extensible Messaging and Presence Protocol (XMPP): Address Format

Abstract

This document defines the address format for the Extensible Messaging and Presence Protocol (XMPP), including support for code points outside the ASCII range. This document obsoletes RFC 6122.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7622.

Table of Contents

1. Introduction

The Extensible Messaging and Presence Protocol (XMPP) [RFC6120] is an application profile of the Extensible Markup Language [XML] for streaming XML data in close to real time between any two or more network-aware entities. The address format for XMPP entities was originally developed in the Jabber open-source community in 1999, first described by [XEP-0029] in 2002, and then defined canonically by [RFC3920] in 2004 and [RFC6122] in 2011.

As specified in RFCs 3920 and 6122, the XMPP address format used the "stringprep" technology for preparation and comparison of non-ASCII characters [RFC3454]. Following the movement of internationalized domain names away from stringprep, this document defines the XMPP address format in a way that no longer depends on stringprep (see the Preparation, Enforcement, and Comparison of Internationalized Strings (PRECIS) problem statement [RFC6885]). Instead, this document builds upon the internationalization framework defined by the IETF's PRECIS working group [RFC7564].

Although every attempt has been made to ensure that the characters allowed in Jabber Identifiers (JIDs) under stringprep are still allowed and handled in the same way under PRECIS, there is no guarantee of strict backward compatibility because of changes in Unicode and the fact that PRECIS handling is based on Unicode properties, not a hardcoded table of characters. Because it is possible that previously valid JIDs might no longer be valid (or previously invalid JIDs might now be valid), operators of XMPP services are advised to perform careful testing before migrating accounts and other data (see Section 6 of [RFC7613] for guidance).

This document obsoletes RFC 6122.

2. Terminology

Many important terms used in this document are defined in [RFC7564], [RFC5890], [RFC6120], [RFC6365], and [Unicode].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Addresses

3.1. Fundamentals

An XMPP entity is anything that can communicate using XMPP. For historical reasons, the network address of an XMPP entity is called a JID. A valid JID is a string of Unicode code points [Unicode], encoded using UTF-8 [RFC3629], and structured as an ordered sequence of localpart, domainpart, and resourcepart, where the first two parts are demarcated by the '@' character used as a separator and the last two parts are similarly demarcated by the '/' character (e.g., <juliet@example.com/balcony>).

The syntax for a JID is defined as follows, using the Augmented Backus-Naur Form (ABNF) as specified in [RFC5234].

```
jid          = [ localpart "@" ] domainpart [ "/" resourcepart ]
localpart    = 1*1023(userbyte)
               ;
               ; a "userbyte" is a byte used to represent a
               ; UTF-8 encoded Unicode code point that can be
               ; contained in a string that conforms to the
               ; UsernameCaseMapped profile of the PRECIS
               ; IdentifierClass defined in RFC 7613
               ;
domainpart   = IP-literal / IPv4address / ifqdn
               ;
               ; the "IPv4address" and "IP-literal" rules are
               ; defined in RFCs 3986 and 6874, respectively,
               ; and the first-match-wins (a.k.a. "greedy")
               ; algorithm described in Appendix B of RFC 3986
               ; applies to the matching process
               ;
ifqdn        = 1*1023(domainbyte)
               ;
               ; a "domainbyte" is a byte used to represent a
               ; UTF-8 encoded Unicode code point that can be
               ; contained in a string that conforms to RFC 5890
               ;
resourcepart = 1*1023(opaquebyte)
               ;
               ; an "opaquebyte" is a byte used to represent a
               ; UTF-8 encoded Unicode code point that can be
               ; contained in a string that conforms to the
               ; OpaqueString profile of the PRECIS
               ; FreeformClass defined in RFC 7613
               ;
```

All JIDs are based on the foregoing structure. However, note that the formal syntax provided above does not capture all of the rules and restrictions that apply to JIDs, which are described below.

Each allowable portion of a JID (localpart, domainpart, and resourcepart) is 1 to 1023 octets in length, resulting in a maximum total size (including the '@' and '/' separators) of 3071 octets.

Implementation Note: The length limits on JIDs and parts of JIDs are based on octets (bytes), not characters. UTF-8 encoding can result in more than one octet per character.

Implementation Note: When dividing a JID into its component parts, an implementation needs to match the separator characters '@' and '/' before applying any transformation algorithms, which might decompose certain Unicode code points to the separator characters.

Implementation Note: Reuse of the IP-literal rule from [RFC6874] implies that IPv6 addresses are enclosed within square brackets (i.e., beginning with '[' and ending with ']'), which was not the case with the definition of the XMPP address format in [RFC3920] but which was changed in [RFC6122]. Also note that the IP-literal rule was updated between [RFC3986] and [RFC6874] to optionally add a zone identifier to any literal address.

This document defines the native format for JIDs; see [RFC5122] for information about the representation of a JID as a Uniform Resource Identifier (URI) [RFC3986] or Internationalized Resource Identifier (IRI) [RFC3987] and the extraction of a JID from an XMPP URI or IRI.

3.2. Domainpart

The domainpart of a JID is the portion that remains once the following parsing steps are taken:

1. Remove any portion from the first '/' character to the end of the string (if there is a '/' character present).

2. Remove any portion from the beginning of the string to the first '@' character (if there is an '@' character present).

This parsing order is important, as illustrated by example 15 in Section 3.5.

The domainpart is the primary identifier and is the only REQUIRED element of a JID (a mere domainpart is a valid JID). Typically, a domainpart identifies the "home" server to which clients connect for XML routing and data management functionality. However, it is not necessary for an XMPP domainpart to identify an entity that provides core XMPP server functionality (e.g., a domainpart can identify an entity such as a multi-user chat service [XEP-0045], a publish-subscribe service [XEP-0060], or a user directory).

The domainpart for every XMPP service MUST be a fully qualified domain name (FQDN), an IPv4 address, an IPv6 address, or an unqualified hostname (i.e., a text label that is resolvable on a local network).

Informational Note: The term "fully qualified domain name" is not well defined. In [RFC1034], it is also called an absolute domain name, and the two terms are associated in [RFC1535]. The earliest use of the term can be found in [RFC1123]. References to those older specifications ought not to be construed as limiting the

characters of a fully qualified domain name to the ASCII range; for example, [RFC5890] mentions that a fully qualified domain name can contain one or more U-labels.

Interoperability Note: Domainparts that are IP addresses might not be accepted by other services for the purpose of server-to-server communication, and domainparts that are unqualified hostnames cannot be used on public networks because they are resolvable only on a local network.

If the domainpart includes a final character considered to be a label separator (dot) by [RFC1034], this character MUST be stripped from the domainpart before the JID of which it is a part is used for the purpose of routing an XML stanza, comparing against another JID, or constructing an XMPP URI or IRI [RFC5122]. In particular, such a character MUST be stripped before any other canonicalization steps are taken.

In general, the content of a domainpart is an Internationalized Domain Name (IDN) as described in the specifications for Internationalized Domain Names in Applications (commonly called "IDNA2008"), and a domainpart is an "IDNA-aware domain name slot" as defined in [RFC5890].

After any and all normalization, conversion, and mapping of code points as well as encoding of the string as UTF-8, a domainpart MUST NOT be zero octets in length and MUST NOT be more than 1023 octets in length. (Naturally, the length limits of [RFC1034] apply, and nothing in this document is to be interpreted as overriding those more fundamental limits.)

Detailed rules and considerations for preparation, enforcement, and comparison are provided in the following sections.

3.2.1. Preparation

An entity that prepares a string for inclusion in an XMPP domainpart slot MUST ensure that the string consists only of Unicode code points that are allowed in NR-LDH labels or U-labels as defined in [RFC5890]. This implies that the string MUST NOT include A-labels as defined in [RFC5890]; each A-label MUST be converted to a U-label during preparation of a string for inclusion in a domainpart slot. In addition, the string MUST be encoded as UTF-8 [RFC3629].

3.2.2. Enforcement

An entity that performs enforcement in XMPP domainpart slots MUST prepare a string as described in Section 3.2.1 and MUST also apply the normalization, case-mapping, and width-mapping rules defined in [RFC5892].

> Informational Note: The order in which the rules are applied for IDNA2008 (see [RFC5892] and [RFC5895]) is different from the order for localparts and resourceparts as described under Sections 3.3 and 3.4.

3.2.3. Comparison

An entity that performs comparison of two strings before or after their inclusion in XMPP domainpart slots MUST prepare each string as specified in Section 3.2.1 and then enforce the normalization, case-mapping, and width-mapping rules specified in Section 3.2.2. The two strings are to be considered equivalent if they are an exact octet-for-octet match (sometimes called "bit-string identity").

3.3. Localpart

The localpart of a JID is an optional identifier placed before the domainpart and separated from the latter by the '@' character. Typically, a localpart uniquely identifies the entity requesting and using network access provided by a server (i.e., a local account), although it can also represent other kinds of entities (e.g., a chatroom associated with a multi-user chat service [XEP-0045]). The entity represented by an XMPP localpart is addressed within the context of a specific domain (i.e., <localpart@domainpart>).

The localpart of a JID MUST NOT be zero octets in length and MUST NOT be more than 1023 octets in length. This rule is to be enforced after any normalization and mapping of code points as well as encoding of the string as UTF-8.

The localpart of a JID is an instance of the UsernameCaseMapped profile of the PRECIS IdentifierClass, which is specified in [RFC7613]. The rules and considerations provided in that specification MUST be applied to XMPP localparts.

> Implementation Note: XMPP uses the Simple Authentication and Security Layer (SASL) [RFC4422] for authentication. At the time of this writing, some SASL mechanisms use SASLprep [RFC4013] for the handling of usernames and passwords; in the future, these SASL mechanisms will likely transition to the use of PRECIS-based handling rules as specified in [RFC7613]. For a detailed

discussion about the implications of that transition (including the potential need to modify or remove certain characters in the underlying account database), see both Section 6 and Appendix A of [RFC7613].

3.3.1. Further Excluded Characters

In XMPP, the following characters are explicitly disallowed in XMPP localparts, even though they are allowed by the IdentifierClass base class and the UsernameCaseMapped profile:

" U+0022 (QUOTATION MARK)

& U+0026 (AMPERSAND)

' U+0027 (APOSTROPHE)

/ U+002F (SOLIDUS)

: U+003A (COLON)

< U+003C (LESS-THAN SIGN)

> U+003E (GREATER-THAN SIGN)

@ U+0040 (COMMERCIAL AT)

Implementation Note: An XMPP-specific method for escaping the foregoing characters (along with U+0020, i.e., ASCII space) has been defined in the JID Escaping specification [XEP-0106].

3.4. Resourcepart

The resourcepart of a JID is an optional identifier placed after the domainpart and separated from the latter by the '/' character. A resourcepart can modify either a <localpart@domainpart> address or a mere <domainpart> address. Typically, a resourcepart uniquely identifies a specific connection (e.g., a device or location) or object (e.g., an occupant in a multi-user chatroom [XEP-0045]) belonging to the entity associated with an XMPP localpart at a domain (i.e., <localpart@domainpart/resourcepart>).

XMPP entities SHOULD consider resourceparts to be opaque strings and SHOULD NOT impute meaning to any given resourcepart. In particular:

- Use of the '/' character as a separator between the domainpart and the resourcepart does not imply that XMPP addresses are hierarchical in the way that, say, HTTP URIs are hierarchical (see [RFC3986] for general discussion); thus, for example, an XMPP address of the form <localpart@domainpart/foo/bar> does not identify a resource "bar" that exists below a resource "foo" in a hierarchy of resources associated with the entity "localpart@domainpart".

- The '@' character is allowed in the resourcepart and is often used in the "handle" shown in XMPP chatrooms [XEP-0045]. For example, the JID <room@chat.example.com/user@host> describes an entity who is an occupant of the room <room@chat.example.com> with a handle of <user@host>. However, chatroom services do not necessarily check such an asserted handle against the occupant's real JID.

The resourcepart of a JID MUST NOT be zero octets in length and MUST NOT be more than 1023 octets in length. This rule is to be enforced after any normalization and mapping of code points as well as encoding of the string as UTF-8.

The resourcepart of a JID is an instance of the OpaqueString profile of the PRECIS FreeformClass, which is specified in [RFC7613]. The rules and considerations provided in that specification MUST be applied to XMPP resourceparts.

3.4.1. Applicability to XMPP Extensions

In some contexts, it might be appropriate to apply more restrictive rules to the preparation, enforcement, and comparison of XMPP resourceparts. For example, in XMPP Multi-User Chat [XEP-0045] it might be appropriate to apply the rules specified in [PRECIS-Nickname]. However, the application of more restrictive rules is out of scope for resourceparts in general and is properly defined in specifications for the relevant XMPP extensions.

3.5. Examples

The following examples illustrate a small number of JIDs that are consistent with the format defined above (note that the characters "<" and ">" are used to delineate the actual JIDs and are not part of the JIDs themselves).

#	JID	Notes
1	<juliet@example.com>	A "bare JID"
2	<juliet@example.com/foo>	A "full JID"
3	<juliet@example.com/foo bar>	Single space in resourcepart
4	<juliet@example.com/foo@bar>	"At" sign in resourcepart
5	<foo\20bar@example.com>	Single space in localpart, as optionally escaped using the XMPP JID Escaping extension
6	<fussball@example.com>	Another bare JID
7	<fußball@example.com>	The third character is LATIN SMALL LETTER SHARP S (U+00DF)
8	<π@example.com>	A localpart of GREEK SMALL LETTER PI (U+03C0)
9	<Σ@example.com/foo>	A localpart of GREEK CAPITAL LETTER SIGMA (U+03A3)
10	<σ@example.com/foo>	A localpart of GREEK SMALL LETTER SIGMA (U+03C3)
11	<ς@example.com/foo>	A localpart of GREEK SMALL LETTER FINAL SIGMA (U+03C2)
12	<king@example.com/&#x265A>;	A resourcepart of the Unicode character BLACK CHESS KING (U+265A)
13	<example.com>	A domainpart
14	<example.com/foobar>	A domainpart and resourcepart
15	<a.example.com/b@example.net>	A domainpart followed by a resourcepart that contains an "at" sign

Table 1: A Sample of Legal JIDs

Several points are worth noting. Regarding examples 6 and 7: although in German the character esszett (LATIN SMALL LETTER SHARP S (U+00DF)) can mostly be used interchangeably with the two characters "ss", the localparts in these examples are different, and (if desired) a server would need to enforce a registration policy that disallows one of them if the other is registered. Regarding examples 9, 10, and 11: case-mapping of GREEK CAPITAL LETTER SIGMA (U+03A3) to lowercase (i.e., to GREEK SMALL LETTER SIGMA (U+03C3)) during comparison would result in matching the JIDs in examples 9 and 10; however, because the PRECIS mapping rules do not account for the special status of GREEK SMALL LETTER FINAL SIGMA (U+03C2), the JIDs in examples 9 and 11 or examples 10 and 11 would not be matched. Regarding example 12: symbol characters such as BLACK CHESS KING (U+265A) are allowed by the PRECIS FreeformClass and thus can be used in resourceparts. Regarding examples 14 and 15: JIDs consisting of a domainpart and resourcepart are rarely seen in the wild but are allowed according to the XMPP address format. Example 15 illustrates the need for careful extraction of the domainpart as described in Section 3.2.

The following examples illustrate strings that are not JIDs because they violate the format defined above.

#	Non-JID string	Notes
16	<"juliet"@example.com>	Quotation marks (U+0022) in localpart
17	<foo bar@example.com>	Space (U+0020) in localpart
18	<juliet@example.com/ foo>	Leading space in resourcepart
19	<@example.com/>	Zero-length localpart and resourcepart
20	<henryⅣ@example.com>	The sixth character is ROMAN NUMERAL FOUR (U+2163)
21	<♚@example.com>	A localpart of BLACK CHESS KING (U+265A)
22	<juliet@>	A localpart without a domainpart
23	</foobar>	A resourcepart without a domainpart

Table 2: A Sample of Strings That Violate the JID Rules

Here again, several points are worth noting. Regarding example 17: even though ASCII space (U+0020) is disallowed in the PRECIS IdentifierClass, it can be escaped to "\20" in XMPP localparts by using the JID Escaping rules defined in [XEP-0106], as illustrated by example 5 in Table 1. Regarding example 20: the Unicode character ROMAN NUMERAL FOUR (U+2163) has a compatibility equivalent of the string formed of LATIN CAPITAL LETTER I (U+0049) and LATIN CAPITAL LETTER V (U+0056), but characters with compatibility equivalents are not allowed in the PRECIS IdentifierClass. Regarding example 21: symbol characters such as BLACK CHESS KING (U+265A) are not allowed in the PRECIS IdentifierClass; however, both of the non-ASCII characters in examples 20 and 21 are allowed in the PRECIS FreeformClass and therefore in the XMPP resourcepart (as illustrated for U+265A by example 12 in Table 1). Regarding examples 22 and 23: the domainpart is required in a JID.

4. Enforcement in JIDs and JID Parts

Enforcement entails applying all of the rules specified in this document. Enforcement of the XMPP address format rules is the responsibility of XMPP servers. Although XMPP clients SHOULD prepare complete JIDs and parts of JIDs in accordance with this document before including them in protocol slots within XML streams, XMPP servers MUST enforce the rules wherever possible and reject stanzas and other XML elements that violate the rules (for stanzas, by returning a <jid-malformed/> error to the sender as described in Section 8.3.3.8 of [RFC6120]).

Entities that enforce the rules specified in this document are encouraged to be liberal in what they accept by following this procedure:

1. Where possible, map characters (e.g., through width mapping, additional mapping, special mapping, case mapping, or normalization) and accept the mapped string.

2. If mapping is not possible (e.g., because a character is disallowed in the FreeformClass), reject the string and return a <jid-malformed/> error.

Enforcement applies to complete JIDs and to parts of JIDs. To facilitate implementation, this document defines the concepts of "JID slot", "localpart slot", and "resourcepart slot" (similar to the concept of a "domain name slot" for IDNA2008 as defined in Section 2.3.2.6 of [RFC5890]):

JID Slot: An XML element or attribute explicitly designated in XMPP or in XMPP extensions for carrying a complete JID.

Localpart Slot: An XML element or attribute explicitly designated in XMPP or in XMPP extensions for carrying the localpart of a JID.

Resourcepart Slot: An XML element or attribute explicitly designated in XMPP or in XMPP extensions for carrying the resourcepart of a JID.

A server is responsible for enforcing the address format rules when receiving protocol elements from clients where the server is expected to handle such elements directly or to use them for purposes of routing a stanza to another domain or delivering a stanza to a local entity; two examples from [RFC6120] are the ’to’ attribute on XML stanzas (which is a JID slot used by XMPP servers for routing of outbound stanzas) and the <resource/> child of the <bind/> element (which is a resourcepart slot used by XMPP servers for binding of a

resource to an account for routing of stanzas between the server and a particular client). An example from [RFC6121] is the 'jid' attribute of the roster <item/> element.

A server is not responsible for enforcing the rules when the protocol elements are intended for communication among other entities, typically within the payload of a stanza that the server is merely routing to another domain or delivering to a local entity. Two examples are the 'initiator' attribute in the Jingle extension [XEP-0166] (which is a JID slot used for client-to-client coordination of multimedia sessions) and the 'nick' attribute in the Multi-User Chat extension [XEP-0045] (which is a resourcepart slot used for administrative purposes in the context of XMPP chatrooms). In such cases, the entities involved SHOULD enforce the rules themselves and not depend on the server to do so, and client implementers need to understand that not enforcing the rules can lead to a degraded user experience or to security vulnerabilities. However, when an add-on service (e.g., a multi-user chat service) handles a stanza directly, it ought to enforce the rules as well, as defined in the relevant specification for that type of service.

This document does not provide an exhaustive list of JID slots, localpart slots, or resourcepart slots. However, implementers of core XMPP servers are advised to consider as JID slots at least the following elements and attributes when they are handled directly or used for purposes of routing to another domain or delivering to a local entity:

- The 'from' and 'to' stream attributes and the 'from' and 'to' stanza attributes [RFC6120].
- The 'jid' attribute of the roster <item/> element for contact list management [RFC6121].
- The 'value' attribute of the <item/> element for Privacy Lists [RFC3921] [XEP-0016] when the value of the 'type' attribute is "jid".
- The 'jid' attribute of the <item/> element for Service Discovery defined in [XEP-0030].
- The <value/> element for Data Forms [XEP-0004] when the 'type' attribute is "jid-single" or "jid-multi".
- The 'jid' attribute of the <conference/> element for Bookmark Storage [XEP-0048].

- The <JABBERID/> of the <vCard/> element for vCard 3.0 [XEP-0054] and the <uri/> child of the <impp/> element for vCard 4.0 [XEP-0292] when the XML character data identifies an XMPP URI [RFC5122].

- The 'from' attribute of the <delay/> element for Delayed Delivery [XEP-0203].

- The 'jid' attribute of the <item/> element for the Blocking Command [XEP-0191].

- The 'from' and 'to' attributes of the <result/> and <verify/> elements for Server Dialback [XEP-0220].

- The 'from' and 'to' attributes of the <iq/>, <message/>, and <presence/> elements for the Jabber Component Protocol [XEP-0114].

Developers of XMPP clients and specialized XMPP add-on services are advised to check the appropriate specifications for JID slots, localpart slots, and resourcepart slots in XMPP protocol extensions such as Service Discovery [XEP-0030], Multi-User Chat [XEP-0045], Publish-Subscribe [XEP-0060], SOCKS5 Bytestreams [XEP-0065], In-Band Registration [XEP-0077], Roster Item Exchange [XEP-0144], and Jingle [XEP-0166].

5. Internationalization Considerations

XMPP applications MUST support IDNA2008 for domainparts as described under Section 3.2, the UsernameCaseMapped profile for localparts as described under Section 3.3, and the OpaqueString profile for resourceparts as described under Section 3.4. This enables XMPP addresses to include a wide variety of characters outside the ASCII range. Rules for enforcement of the XMPP address format are provided in [RFC6120] and specifications for various XMPP extensions.

> Interoperability Note: For backward compatibility, many existing XMPP implementations and deployments support IDNA2003 [RFC3490] for domainparts, and the stringprep [RFC3454] profiles Nodeprep and Resourceprep [RFC3920] for localparts and resourceparts.

6. IANA Considerations

6.1. Stringprep Profiles Registry

The stringprep specification [RFC3454] did not provide for entries in the "Stringprep Profiles" registry to have any state except "Current" or "Not Current". Because this document obsoletes RFC 6122, which registered the Nodeprep and Resourceprep profiles of stringprep, IANA has marked those profiles as "Not Current" and cited this document as an additional reference.

7. Security Considerations

7.1. Reuse of PRECIS

The security considerations described in [RFC7564] apply to the IdentifierClass and FreeformClass base string classes used in this document for XMPP localparts and resourceparts, respectively. The security considerations described in [RFC5890] apply to internationalized domain names, which are used here for XMPP domainparts.

7.2. Reuse of Unicode

The security considerations described in [UTS39] apply to the use of Unicode characters in XMPP addresses.

7.3. Address Spoofing

There are two forms of address spoofing: forging and mimicking.

7.3.1. Address Forging

In the context of XMPP technologies, address forging occurs 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 (or an authorization identity provided during negotiation of SASL authentication [RFC4422] as described in [RFC6120]). For example, address forging occurs if an entity that authenticated as "juliet@im.example.com" is able to send XML stanzas from "nurse@im.example.com" or "romeo@example.net".

Address forging is difficult in XMPP systems, given the requirement for sending servers to stamp 'from' addresses and for receiving servers to verify sending domains via server-to-server authentication (see [RFC6120]). However, address forging is possible if:

- A poorly implemented server ignores the requirement for stamping the 'from' address. This would enable any entity that authenticated with the server to send stanzas from any localpart@domainpart as long as the domainpart matches the sending domain of the server.

- An actively malicious server generates stanzas on behalf of any registered account at the domain or domains hosted at that server.

Therefore, an entity outside the security perimeter of a particular server cannot reliably distinguish between JIDs of the form <localpart@domainpart> at that server and thus can authenticate only the domainpart of such JIDs with any level of assurance. This specification does not define methods for discovering or counteracting the kind of poorly implemented or rogue servers just described. However, the end-to-end authentication or signing of XMPP stanzas could help to mitigate this risk, because it would require the rogue server to generate false credentials for signing or encryption of each stanza, in addition to modifying 'from' addresses.

7.3.2. Address Mimicking

Address mimicking occurs when an entity provides legitimate authentication credentials for, and sends XML stanzas from, an account whose JID appears to a human user to be the same as another JID. Because many characters are visually similar, it is relatively easy to mimic JIDs in XMPP systems. As one simple example, the localpart "ju1iet" (using the Arabic numeral one as the third character) might appear the same as the localpart "juliet" (using lowercase "L" as the third character).

As explained in [RFC5890], [RFC7564], [UTR36], and [UTS39], there is no straightforward solution to the problem of visually similar characters. Furthermore, IDNA and PRECIS technologies do not attempt to define such a solution. As a result, XMPP domainparts, localparts, and resourceparts could contain such characters, leading to security vulnerabilities such as the following:

- A domainpart is always employed as one part of an entity's address in XMPP. One common usage is as the address of a server or server-side service, such as a multi-user chat service [XEP-0045]. The security of such services could be compromised based on different interpretations of the internationalized domainpart; for

example, a user might authorize a malicious entity at a fake server to view the user's presence information, or a user could join chatrooms at a fake multi-user chat service.

- A localpart can be employed as one part of an entity's address in XMPP. One common usage is as the username of an instant messaging user; another is as the name of a multi-user chatroom; and many other kinds of entities could use localparts as part of their addresses. The security of such services could be compromised based on different interpretations of the internationalized localpart; for example, a user entering a single internationalized localpart could access another user's account information, or a user could gain access to a hidden or otherwise restricted chatroom or service.

- A resourcepart can be employed as one part of an entity's address in XMPP. One common usage is as the name for an instant messaging user's connected resource; another is as the nickname of a user in a multi-user chatroom; and many other kinds of entities could use resourceparts as part of their addresses. The security of such services could be compromised based on different interpretations of the internationalized resourcepart; for example, two or more confusable resources could be bound at the same time to the same account (resulting in inconsistent authorization decisions in an XMPP application that uses full JIDs), or a user could send a private message to someone other than the intended recipient in a multi-user chatroom.

XMPP services and clients are strongly encouraged to define and implement consistent policies regarding the registration, storage, and presentation of visually similar characters in XMPP systems. In particular, service providers and software implementers are strongly encouraged to apply the policies recommended in [RFC7564].

8. Conformance Requirements

This section describes a protocol feature set that summarizes the conformance requirements of this specification (similar feature sets are provided for XMPP in [RFC6120] and [RFC6121]). The summary is purely informational, and the conformance keywords of [RFC2119] as used here are intended only to briefly describe the referenced normative text from the body of this specification. This feature set is appropriate for use in software certification, interoperability testing, and implementation reports. For each feature, this section provides the following information:

- A human-readable name
- An informational description
- A reference to the particular section of this document that normatively defines the feature
- Whether the feature applies to the client role, the server role, or both (where "N/A" signifies that the feature is not applicable to the specified role)
- Whether the feature MUST or SHOULD be implemented, where the capitalized terms are to be understood as described in [RFC2119]

The feature set specified here provides a basis for interoperability testing and follows the spirit of a proposal made by Larry Masinter within the IETF's NEWTRK working group in 2005 [INTEROP].

Feature: address-domain-length

Description: Ensure that the domainpart of an XMPP address is at least one octet in length and at most 1023 octets in length, and that it conforms to the underlying length limits of the DNS.

Section: Section 3.2

Roles: Server MUST, client SHOULD.

Feature: address-domain-prep

Description: Ensure that the domainpart of an XMPP address conforms to IDNA2008, that it contains only NR-LDH labels and U-labels (not A-labels), and that all uppercase and titlecase code points are mapped to their lowercase equivalents.

Section: Section 3.2

Roles: Server MUST, client SHOULD.

Feature: address-localpart-length

Description: Ensure that the localpart of an XMPP address is at least one octet in length and at most 1023 octets in length.

Section: Section 3.3

Roles: Server MUST, client SHOULD.

Feature: address-localpart-prep

Description: Ensure that the localpart of an XMPP address conforms to the UsernameCaseMapped profile of the PRECIS IdentifierClass.

Section: Section 3.3

Roles: Server MUST, client SHOULD.

Feature: address-resource-length

Description: Ensure that the resourcepart of an XMPP address is at least one octet in length and at most 1023 octets in length.

Section: Section 3.4

Roles: Server MUST, client SHOULD.

Feature: address-resource-prep

Description: Ensure that the resourcepart of an XMPP address conforms to the OpaqueString profile of the PRECIS FreeformClass.

Section: Section 3.4

Roles: Server MUST, client SHOULD.

9. References

9.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <http://www.rfc-editor.org/info/rfc3986>.

[RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <http://www.rfc-editor.org/info/rfc5890>.

[RFC5892] Faltstrom, P., Ed., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, DOI 10.17487/RFC5892, August 2010, <http://www.rfc-editor.org/info/rfc5892>.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <http://www.rfc-editor.org/info/rfc6120>.

[RFC6365] Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, DOI 10.17487/RFC6365, September 2011, <http://www.rfc-editor.org/info/rfc6365>.

[RFC6874] Carpenter, B., Cheshire, S., and R. Hinden, "Representing IPv6 Zone Identifiers in Address Literals and Uniform Resource Identifiers", RFC 6874, DOI 10.17487/RFC6874, February 2013, <http://www.rfc-editor.org/info/rfc6874>.

[RFC7564] Saint-Andre, P. and M. Blanchet, "PRECIS Framework: Preparation, Enforcement, and Comparison of Internationalized Strings in Application Protocols", RFC 7564, DOI 10.17487/RFC7564, May 2015, <http://www.rfc-editor.org/info/rfc7564>.

[RFC7613] Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", RFC 7613, DOI 10.17487/RFC7613, August 2015, <http://www.rfc-editor.org/info/rfc7613>.

[Unicode] The Unicode Consortium, "The Unicode Standard", <http://www.unicode.org/versions/latest/>.

[UTR36] Unicode Technical Report #36, "Unicode Security Considerations", edited by Mark Davis and Michel Suignard, <http://www.unicode.org/reports/tr36/>.

9.2. Informative References

[INTEROP] Masinter, L., "Formalizing IETF Interoperability Reporting", Work in Progress, draft-ietf-newtrk-interop-reports-00, October 2005.

[PRECIS-Nickname] Saint-Andre, P., "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Nicknames", Work in Progress, draft-ietf-precis-nickname-18, June 2015.

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, DOI 10.17487/RFC1123, October 1989, <http://www.rfc-editor.org/info/rfc1123>.

[RFC1535] Gavron, E., "A Security Problem and Proposed Correction With Widely Deployed DNS Software", RFC 1535, DOI 10.17487/RFC1535, October 1993, <http://www.rfc-editor.org/info/rfc1535>.

[RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, DOI 10.17487/RFC3454, December 2002, <http://www.rfc-editor.org/info/rfc3454>.

[RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, DOI 10.17487/RFC3490, March 2003, <http://www.rfc-editor.org/info/rfc3490>.

[RFC3920] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, DOI 10.17487/RFC3920, October 2004, <http://www.rfc-editor.org/info/rfc3920>.

[RFC3921] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 3921, DOI 10.17487/RFC3921, October 2004, <http://www.rfc-editor.org/info/rfc3921>.

[RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, DOI 10.17487/RFC3987, January 2005, <http://www.rfc-editor.org/info/rfc3987>.

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, DOI 10.17487/RFC4013, February 2005, <http://www.rfc-editor.org/info/rfc4013>.

[RFC4422] Melnikov, A., Ed., and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", RFC 4422, DOI 10.17487/RFC4422, June 2006, <http://www.rfc-editor.org/info/rfc4422>.

[RFC5122] Saint-Andre, P., "Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP)", RFC 5122, DOI 10.17487/RFC5122, February 2008, <http://www.rfc-editor.org/info/rfc5122>.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, DOI 10.17487/RFC5895, September 2010, <http://www.rfc-editor.org/info/rfc5895>.

[RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, DOI 10.17487/RFC6121, March 2011, <http://www.rfc-editor.org/info/rfc6121>.

[RFC6122] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Address Format", RFC 6122, DOI 10.17487/RFC6122, March 2011, <http://www.rfc-editor.org/info/rfc6122>.

[RFC6885] Blanchet, M. and A. Sullivan, "Stringprep Revision and Problem Statement for the Preparation and Comparison of Internationalized Strings (PRECIS)", RFC 6885, DOI 10.17487/RFC6885, March 2013, <http://www.rfc-editor.org/info/rfc6885>.

[UTS39] Unicode Technical Standard #39, "Unicode Security Mechanisms", edited by Mark Davis and Michel Suignard, <http://unicode.org/reports/tr39/>.

[XEP-0004] Eatmon, R., Hildebrand, J., Miller, J., Muldowney, T., and P. Saint-Andre, "Data Forms", XSF XEP 0004, August 2007, <http://xmpp.org/extensions/xep-0004.html>.

[XEP-0016] Millard, P. and P. Saint-Andre, "Privacy Lists", XSF XEP 0016, February 2007, <http://xmpp.org/extensions/xep-0016.html>.

[XEP-0029] Kaes, C., "Definition of Jabber Identifiers (JIDs)", XSF XEP 0029, October 2003, <http://xmpp.org/extensions/xep-0029.html>.

[XEP-0030] Hildebrand, J., Millard, P., Eatmon, R., and P. Saint-Andre, "Service Discovery", XSF XEP 0030, June 2008, <http://xmpp.org/extensions/xep-0030.html>.

[XEP-0045] Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, February 2012, <http://xmpp.org/extensions/xep-0045.html>.

[XEP-0048] Blackman, R., Millard, P., and P. Saint-Andre, "Bookmarks", XSF XEP 0048, November 2007, <http://xmpp.org/extensions/xep-0048.html>.

[XEP-0054] Saint-Andre, P., "vcard-temp", XSF XEP 0054, July 2008, <http://xmpp.org/extensions/xep-0054.html>.

[XEP-0060] Millard, P., Saint-Andre, P., and R. Meijer, "Publish-Subscribe", XSF XEP 0060, July 2010, <http://xmpp.org/extensions/xep-0060.html>.

[XEP-0065] Smith, D., Miller, M., Saint-Andre, P., and J. Karneges, "SOCKS5 Bytestreams", XSF XEP 0065, April 2011, <http://xmpp.org/extensions/xep-0065.html>.

[XEP-0077] Saint-Andre, P., "In-Band Registration", XSF XEP 0077, January 2012, <http://xmpp.org/extensions/xep-0077.html>.

[XEP-0106] Hildebrand, J. and P. Saint-Andre, "JID Escaping", XSF XEP 0106, June 2007, <http://xmpp.org/extensions/xep-0106.html>.

[XEP-0114] Saint-Andre, P., "Jabber Component Protocol", XSF XEP 0114, January 2012, <http://xmpp.org/extensions/xep-0114.html>.

[XEP-0144] Saint-Andre, P., "Roster Item Exchange", XSF XEP 0144, August 2005, <http://xmpp.org/extensions/xep-0144.html>.

[XEP-0166] Ludwig, S., Beda, J., Saint-Andre, P., McQueen, R., Egan, S., and J. Hildebrand, "Jingle", XSF XEP 0166, December 2009, <http://xmpp.org/extensions/xep-0166.html>.

[XEP-0191] Saint-Andre, P., "Blocking Command", XSF XEP 0191, July 2012, <http://xmpp.org/extensions/xep-0191.html>.

[XEP-0203] Saint-Andre, P., "Delayed Delivery", XSF XEP 0203, September 2009, <http://xmpp.org/extensions/xep-0203.html>.

[XEP-0220] Miller, J., Saint-Andre, P., and P. Hancke, "Server Dialback", XSF XEP 0220, August 2014, <http://xmpp.org/extensions/xep-0220.html>.

[XEP-0292] Saint-Andre, P. and S. Mizzi, "vCard4 Over XMPP", XSF XEP 0292, September 2013, <http://xmpp.org/extensions/xep-0292.html>.

[XML] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126>.

Appendix A. Differences from RFC 6122

Based on consensus derived from working group discussion, implementation and deployment experience, and formal interoperability testing, the following substantive modifications were made from RFC 6122.

- Changed domainpart preparation to use IDNA2008 (instead of IDNA2003).
- Changed localpart preparation to use the UsernameCaseMapped profile of the PRECIS IdentifierClass (instead of the Nodeprep profile of stringprep).
- Changed resourcepart preparation to use the OpaqueString profile of the PRECIS FreeformClass (instead of the Resourceprep profile of stringprep).
- Specified that internationalized labels within domainparts must be U-labels (instead of "should be" U-labels).
- Specified that fullwidth and halfwidth characters must be mapped to their decomposition mappings (previously handled through the use of Normalization Form KC).
- Specified the use of Unicode Normalization Form C (instead of Unicode Normalization Form KC as specified in the Nodeprep and Resourceprep profiles of stringprep).
- Specified that servers must enforce the address-formatting rules.

Acknowledgements

Thanks to Ben Campbell, Dave Cridland, Miguel Garcia, Joe Hildebrand, Jonathan Lennox, Matt Miller, Florian Schmaus, Sam Whited, and Florian Zeitz for their input during working group discussion.

Dan Romascanu completed a helpful review on behalf of the General Area Review Team.

During IESG review, Alissa Cooper, Brian Haberman, and Barry Leiba provided comments that led to improvements in the document.

Thanks also to Matt Miller in his role as document shepherd, Joe Hildebrand in his role as working group chair, and Ben Campbell in his role as sponsoring Area Director.

The author wishes to acknowledge Cisco Systems, Inc., for employing him during his work on earlier draft versions of this document.

Author’s Address

Peter Saint-Andre
&yet

Email: peter@andyet.com
URI: https://andyet.com/