INTERNET-DRAFT There are 5 ISSUES highlighted like this.
<draft-ietf-ipp-not-~~http~~ntfy-delivery-00.txt>

Hugo Parra
Novell, Inc.
Tom Hastings
Xerox Corp.

~~October 19~~December 9, 1999

Internet Printing Protocol/1.1: ~~HTTP-Based IPP~~ The 'ipp-ntfy' Notification Delivery Method and Protocol

ISSUE 01 - What should the name of this delivery method and protocol be that we use in the title of this document?

ISSUE 02 - What should the scheme name be? Consider 'ipp-ntfy' a working title, until we see several schemes. The 'ipp-get' delivery method is another example. Should the scheme name somehow include "notification", i.e., 'ntfy'? How about 'ipp-ntfy-send' or 'ipp-ntfy-push' and 'ipp-ntfy-get' or 'ipp-ntfy-pull' to go with the Send-Notifications and Get-Notifications operations, respectively?

ISSUE 03 - Should the scheme name be used in the title?

Abstract

The IPP event notification specification [ipp-ntfy] is an OPTIONAL extension to IPP/1.0 and IPP/1.1. [ipp-ntfy] requires the ~~availability~~ definition of one or more delivery methods for dispatching event notification reports to ~~interested parties~~Notification Recipients. This document describes the semantics and syntax of ~~a protocol that a~~ the 'ipp-ntfy' event notification delivery method that is itself a request/response protocol. For this delivery method, an IPP Printer ~~may use to deliver~~ sends (pushes) IPP event Notifications to the Notification Recipients using the protocol defined herein which includes HTTP ~~for~~ as a transport.

The full set of IPP documents includes:

Design Goals for an Internet Printing Protocol [RFC2567]
Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
Internet Printing Protocol/1.1: Model and Semantics (this document)
Internet Printing Protocol/1.1: Encoding and Transport [ipp-pro]
Internet Printing Protocol/1.1: Implementer's Guide [ipp-iig]
Mapping between LPD and IPP Protocols [RFC2569]

The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included in a printing protocol for the Internet. It identifies requirements for three types of users: end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0. A few OPTIONAL operator operations have been added to IPP/1.1.

The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document describes IPP from a high level view, defines a roadmap for the various documents that form the suite of IPP specification documents, and gives background and rationale for the IETF working group's major decisions.

The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It defines the encoding rules for a new Internet MIME media type called "application/ipp". This document also defines the rules for transporting ~~over HTTP~~ a message body over HTTP whose Content-Type is "application/ipp". This document defines a new scheme named 'ipp' for identifying IPP printers and jobs.

The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some of the considerations that may assist them in the design of their client and/or IPP object implementations. For example, a typical order of processing requests is given, including error checking. Motivation for some of the specification decisions is also included.

The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of gateways between IPP and LPD (Line Printer Daemon) implementations.

Table of Contents

1 Introduction

IPP ~~p~~Printers that support the OPTIONAL IPP event notification extension [ipp-ntfy] either a) accept, store, and use notification Subscriptions to generate notification reports and implement one or more delivery methods for notifying interested parties, or b) support a subset of these tasks and farm out the remaining tasks to a Notification Delivery Service. The ~~protocol~~ 'ipp-ntfy' event notification delivery method specified in this document is itself a request/response protocol that may be used in a variety of notification scenarios. Its primary intended use is for IPP ~~p~~Printers to send (push) event notifications to ~~n~~Notification ~~r~~Recipients using the IPP Send-Notifications operation over HTTP. However, it may also be used by IPP ~~p~~Printers to send notification to Notification Services and by Notification Delivery Services to send notifications to the Ultimate ~~n~~Notification ~~r~~Recipients (see [ipp-ntfy]). Furthermore, this protocol can be extended in the future to add other operations, such as querying a Notification Recipient for its capabilities.

2 Terminology

This document uses terms such as "attributes", "keywords", and "support". These terms have special meaning and are defined in the model terminology [ipp-mod] section 12.2.

Capitalized terms, such as MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, and OPTIONAL, have special meaning relating to conformance. These terms are defined in [ipp-mod] section 12.1 on conformance terminology, most of which is taken from RFC 2119 [RFC2119].

This section defines the following additional terms that are used throughout this document:

REQUIRED: if an implementation supports the extensions described in this document, it MUST support a REQUIRED feature.
OPTIONAL: if an implementation supports the extensions described in this document, it MAY support an OPTIONAL feature.
Event Notification (Notification for short) - See [ip-ntfy]
Notification Source - See [ipp-ntfy]
Notification Recipient - See [ipp-ntfy]
Subscription object - See [ipp-ntfy]
Ultimate Notification Recipient - See [ipp-ntfy]

3 Model and Operation

In the IPP Notification Model [ipp-ntfy], one or more Per-Job Subscriptions can be supplied in the Job Creation operation or OPTIONALLY as subsequent Create-Job-Subscription operations; one Per-Printer Subscription can be supplied in the Create-Printer operation. The client that creates these Subscription objects becomes the owner of the Subscription object.

When creating each Subscription object, the client supplies the "notify-recipient" (uri) attribute. The "notify-recipient" attribute specifies both a single Notification Recipient that is to receive the Notifications when subsequent events occur and the method for notification delivery that the IPP Printer is to use. For the Notification delivery method defined in this document, the notification method is 'ipp-ntfy' and the rest of the URI is the address of the Notification Recipient to which the IPP Printer will send the Send-Notifications operations using HTTP as a transport.

The 'ipp-ntfy' event notification delivery method defined in this document is also ~~HTTP-Based IPP Notification Protocol, hereafter referred to as HTTP notification protocol, is~~ a client/server protocol. The "client" in this HTTP relationship is the Notification Source described in [ipp-ntfy] while the "server" is the Notification Recipient. The Notification Source invokes the Send-Notifications operation~~s~~ supported by the 'ipp-ntfy' ~~HTTP~~ notification protocol to communicate IPP event Notification contents to the Notification Recipient. The Notification Recipient only conveys information to the Notification Source in the form of responses to the operations initiated by the Notification Source.

~~HTTP notification~~All requests defined for this protocol will be issued as HTTP POST operations and their corresponding HTTP notification responses will be returned in the responses to those HTTP POST operations. Hence, Notification Sources that implement the ~~HTTP notification~~ 'ipp-ntfy' delivery method and protocol will need to include an HTTP client stack while notification recipients that implement this protocol will need to support an HTTP server stack (see section ~~4~~6 for more details).

~~3.1~~4 ~~HTTP~~ Notification Operations

The ~~job of an HTTP~~ Notification Source ~~is to use the contents of an IPP Notification as defined in [ipp-ntfy] to~~ composes the information defined for an IPP Notification [ipp-ntfy] and sends it using the Sent-Notifications operation to the ~~invoke the appropriate HTTP notification operation and send it to the specified HTTP~~ Notification Recipient supplied in the Subscription object.

The ~~HTTP~~ 'ipp-ntfy' delivery method and notification protocol makes extensive use of the operations model defined by IPP [rfc2566]. This includes, the use of a URI as the identifier for the target of each operation, the inclusion of a version number, operation-id, and request-id in each request, and the definition of attribute groups. The ~~HTTP notification protocol~~ Send-Notifications operation uses the Operation Attributes group, but currently has no need for the Unsupported Attributes, Printer Object Attributes, and Job-Object Attributes groups. However, it ~~defines~~ uses a new attribute group, the ~~Notification~~ Generic Attributes group.

ISSUE 04 - Ok to add a "Generic Attributes" group tag to [ipp-pro], instead of adding a special tag for each new object and/or operation that needs a different set of attributes than Job or Printer? The same issue for the Subscription object in [ipp-ntfy]. Either we define separate tags for both or use a single generic tag for both and future objects and attribute groups.

In its 1.0 version, the ~~HTTP~~ 'ipp-ntfy' delivery method and notification protocol is composed of a single operation, but may be extended in the future as needed (e.g., to find out specific capabilities of an ~~HTTP~~ 'ipp-ntfy' Notification ~~listener~~Recipient). The operation currently defined is Send-Notifications.

~~4.1.1~~4.1 ~~Report-Ipp~~Send-Notifications Operation

This REQUIRED operation allows a Notification Source to send one or more Notifications to a Notification Recipient using HTTP. The operation has been tailored to accommodate the current definition of IPP Notification [ipp-ntfy].

Both ~~'machine-consumable'~~Machine-Consumable and ~~'human-consumable'~~Human-Consumable notifications may be sent to ~~an HTTP~~ a Notification Recipient through this operation.

~~3.1.1.1~~4.1.1 Send-Notifications Request

The following groups of attributes are part of the Send-Notifications Request:

Group 1: Operation Attributes

Natural Language and Character Set:

The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

Target:

The URI of the ~~HTTP~~'ipp-ntfy' Notification Recipient.

Group 2 to N: Notification Attributes

"human-readable-report" (text)

The ~~HTTP~~ 'ipp-ntfy' Notification Source OPTIONALLY ~~supplies~~ supports this attribute. ~~A~~This attribute is a text string generated by the IPP printer or Notification Delivery Service from the contents of the IPP Notification suitable for human consumption. If the Notification Source supports this attribute, it MUST supply this attribute if the Subscription object contains the "notify-text-format" (mimeMediaType) attribute. The text value of this attribute MUST be localized in the charset identified by the "notify-charset" (charset) attribute and the natural language identified by the notify-natural-language" (naturalLanguage) attribute supplied in the associated Subscription object that generates this event Notification. The format of the text value is specified by the value of the "notify-text-format" (mimeMediaType) supplied in the associated Subscription object.

ISSUE 5 - Ok to extend Notification Model to allow a single notification to have both Human Consumable form and Machine Consumable form when the client asks for Human Consumable form by supplying the "notify-text-format" attribute rather than the Human Consumable being sent instead or in addition to the Machine Consumable using MIME multi-part-related?

All of the REQUIRED attributes and any of the OPTIONAL attributes indicated in [ipp-ntfy] for a Push event Notification, including "notify-text-format-type" (mimeMediaType), if the "human-readable-report" (text) attribute is included, so that the Notification Recipient will know the text format of the "human-readable-report" (text) attribute value.

~~"version-number" (integer (0:32767))~~
~~"status-code" (integer (0:32767))~~
~~"request-id" (integer (0:MAX))~~
~~"attributes-charset" (charset)~~
~~"attributes-natural-language" (naturalLanguage)~~
~~"printer-uri" (uri)~~
~~"printer-name" (name(127))~~
~~"job-id" (integer(1:MAX))~~
~~"job-name" (name(MAX))~~
~~"trigger-event" (type2 keyword)~~
~~"trigger-time" (integer(MIN:MAX))~~
~~"trigger-date-time" (dateTime)~~
~~"subscription-id" (integer(1:MAX))~~
~~"subscriber-user-name" (name(MAX))~~
~~"subscriber-user-data" (octetString(63))~~
~~"job-state" (type1 enum)~~
~~"job-state-reasons" (1setOf type2 keyword)~~
~~"job-k-octets-processed" (integer(0:MAX))~~
~~"job-impressions-completed" (integer(0:MAX))~~
~~"job-media-sheets-completed" (integer(0:MAX))~~
~~"job-collation-type" (type2 enum)~~
~~"sheet-completed-copy-number" (integer(-2:MAX))~~
~~"sheet-completed-document-number" (integer(-2:MAX))~~
~~"impressions-interpreted" (integer(-2:MAX))~~
~~"impressions-completed-current-copy" (integer(-2:MAX))~~
~~"printer-state" (type1 enum)~~
~~"printer-state-reasons" (1setOf type2 keyword)~~
~~"printer-is-accepting-jobs" (boolean)~~

These attributes communicate the same information as the notification attributes by the same name described in sections 7.4, 7.5, and 7.6 of [ipp-ntfy]. The rules that govern when each individual attribute MUST or MAY be included in this operation precisely mirror those specified in [ipp-ntfy].

~~3.1.1.2~~4.1.2 Send-Notifications Response

The ~~HTTP~~'ipp-ntfy' Notification Recipient returns a status code for the entire operation and one for each Notification Report in the request if the operation's status code is other than "success-ok". If the ~~HTTP~~'ipp-ntfy' notification listener receives a Notification report that it can't pair up with a subscription it knows about, it can return an error status-code to indicate that events associated with that subscription should no longer be sent to it.

Group 1: Operation Attributes

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes ads defined in [rfc 2566] section 3.1.4.1.

Group 2 to N: Notification Attributes

"notification-report-status-code" (type2 enum)
Indicates whether the ~~HTTP~~'ipp-ntfy' Notification ~~listener~~ Recipient was able to consume the n-th Notification Report.

4.2 ~~HTTP~~ Notification Protocol URI Scheme

The 'ipp-ntfy' event notification delivery method uses the 'ipp-ntfy' URI scheme in the "notify-recipients" attribute in the Subscription object in order to indicate the event notification delivery method defined in this document. The remainder of the URI indicates the host and address of the Notification Recipient that is to receive the Send-Notification operation.

REPEAT OF ISSUE 02 - What should the scheme name be? Consider 'ipp-ntfy' a working title, until we see several schemes. The 'ipp-get' delivery method is another example. Should the scheme name somehow include "notification", i.e., 'ntfy'? How about 'ipp-ntfy-send' or 'ipp-ntfy-push' and 'ipp-ntfy-get' or 'ipp-ntfy-pull' to go with the Send-Notifications and Get-Notifications operations, respectively?

~~ISSUE 2 - *Should the URI scheme for this protocol be "http://", "ipp://", or something else like "ipp-ntfy://". If we intent this proposal to go to the IESG, something along the lines of the third option might be our only alternative*~~

5 Encoding of the Operation Layer

The ~~HTTP~~ 'ipp-ntfy' event notification delivery method and protocol uses the same operation layer encoding model and syntax as IPP [ipp-pro] with two extensions:

5.1 New attribute tag:

a) A new attribute tag is defined:

~~notification-report~~generic-attributes-tag = %x07 ; tag of 7

5.2 New status codes:

b) The following status codes are defined:

5.2.1 ~~0xYYYY -~~ unknown-notification-recipient. (0xYYYY)

The Notification Recipient returns this status code in order to indicate that the intended Ultimate Notification Recipient is not known to the Notification Recipient.

5.2.2 ~~0xZZZZ -~~ unable-to-delivery-notification-report (0xZZZZ)

The Notification Recipient returns this status code in ordre to indicate that it was unable to deliver the event Notification to the intended Ultimate Notification Recipient.

5.2.3 successful-ok-but-cancel-subscription (0xXXXX)

The Notification Recipient indicates that it no longer wants to receive Notifications for this Subscription object. Therefore, the Subscription object is canceled. Note: this status code allows the Notification Recipient to cancel a Subscription object without having to be the owner of the Subscription object. Only the owner of the Subscription object can cancel a Subscription object using the Cancel-Subscription operation.

> ~~ISSUE 3 - Should we add a success status code, say, 'successful-ok-but-cancel-subscription' which requests that the subscription be canceled. Then the Notification Recipient can cancel a subscription that another party established even though the Notification Recipient is not the owner of the Subscription.~~

The encoding for the ~~Report-IPP-~~Send-Notification Request consists of:

```
     -----------------------------------
     |        version-number           | 2 byte
     -----------------------------------
     |         operation-id            | 2 bytes
     -----------------------------------
     |          request-id             | 4 bytes
     -----------------------------------
     |     operation-attributes-tag    | 1 byte
     -----------------------------------
     |    natural-language-attribute   | u bytes
     -----------------------------------
     |       charset-attribute         | v bytes
     -----------------------------------
     |        target-attribute         |       w bytes
     --------------------------------------------
     |        ~~notification~~generic-attributes-tag          | 1 byte  |
     -----------------------------------         | - 1 or more
     |      notification-attr-list     | x bytes |
     --------------------------------------------
     |      end-of-attributes-tag      | 1 byte
     -----------------------------------
```

Where:

version-number is made up of a major-version-number of %d1 and a minor-version-number of %d0 indicating the 1.0 version of the 'ipp-ntfy' event ~~HTTP~~ notification delivery method and protocol.

operation-id, in the 1.0 version of the protocol, can only be 0x00003, ~~Report IPP~~Send-Notification.

request-id is any 4 byte number provided by the notification source and must be matched by the notification recipient in the corresponding response to a request. It assists the notification source in associating operation responses with their corresponding requests. Note that this request id is independent of the request id embedded in the notification report, which is opaque to the delivery method but assists the notification recipient order and identity missing or duplicate notification reports.

operation-attribute tag, natural-language-attribute, charset-attribute, target-attribute, and end-of-attributes-tag have the same syntax and semantics as in [ipp-pro].

notification-attr-list contains a list of the attributes that make up a single notification (see section 2 above) encoded using the syntax specified in [ipp-pro].

The encoding for the Send-Notification Response consists of:

```
     -----------------------------------
     |         version-number          | 2 byte
     -----------------------------------
     |          status-code            | 2 bytes
     -----------------------------------
     |          request-id             | 4 bytes
     -----------------------------------             \
     |    operation-attributes-tag     | 1 byte       |
     -----------------------------------              |
     |  natural-language-attribute     | u bytes      | ~~Not needed in 1.0~~
     -----------------------------------               > ~~<ISSUE 4: Do we~~
     |       charset-attribute         | v bytes      | ~~want to keep it?>~~
     -----------------------------------              |
     |        target-attribute         | w bytes      |
     ----------------------------------------------- /
     |         ~~notification~~generic-attributes-tag      | 1 byte    |
     -----------------------------------          | - 1 or more
     |        ntfy-status-code         | 2 bytes  |
     ----------------------------------------------
     |      end-of-attributes-tag      | 1 byte
     -----------------------------------
```

6 Encoding of Transport Layer

HTTP/1.1 [rfc2616~~rfc2068~~] is the transport layer for this protocol.

The operation layer has been designed with the assumption that the transport layer contains the following information:

- the URI of the target job or printer operation.

- the total length of the data in the operation layer, either as a single length or as a sequence of chunks each with a length.

It is REQUIRED that an ~~HTTP~~ 'ipp-ntfy' Notification Recipient implementation support HTTP over the IANA assigned Well Known Port XXX (the ~~HTTP~~ 'ipp-ntfy' notification protocol default port), though a notification recipient implementation MAY support HTTP over some other port as well.

Each HTTP operation MUST use the POST method where the request-URI is the object target of the operation, and where the "Content-Type" of the message-body in each request and response MUST be "application/ipp-ntfy". The message-body MUST contain the operation layer and MUST have the syntax described in section 3, "Encoding of Operation Layer". An 'ipp-ntfy' ~~HTTP~~ Notification Source implementation MUST adhere to the rules for a client described for HTTP1.1 [rfc2616~~rfc2068~~]. An 'ipp-ntfy' ~~HTTP~~ Notification Recipient implementation MUST adhere the rules for an origin server described for HTTP1.1 [rfc2616~~rfc2068~~].

An 'ipp-ntfy' ~~HTTP~~ Notification Source sends a response for each request that it receives. If a notification recipient detects an error, it MAY send a response before it has read the entire request. If the HTTP layer of the Notification Recipient completes processing the HTTP headers successfully, it MAY send an intermediate response, such as "100 Continue", with no notification data before sending the notification response. The 'ipp-ntfy' ~~HTTP~~ Notification Sources MUST expect such a variety of responses from notification recipients. For further information on HTTP/1.1, consult the HTTP documents [rfc2616~~rfc2068~~].

An 'ipp-ntfy' ~~HTTP~~ Notification Recipient (server) MUST support chunking for HTTP notification requests, and an 'ipp-ntfy' ~~HTTP~~ Notification Source (client) MUST support chunking for HTTP notification responses according to HTTP/1.1[rfc2616~~rfc2068~~]. Note: this rule causes a conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for POST methods, and this rule may cause a conflict with non-compliant implementations of HTTP/1.1 that don't support chunking for CGI scripts

7 IANA Considerations

IANA will be asked to register this 'ipp-ntfy' ~~HTTP~~ notification delivery scheme and protocol and will be asked to assign a default port.

8 Internationalization Considerations

When the client requests Human Consumable form by supplying the "notify-text-format" operation attribute (see [ipp-ntfy]), the IPP Printer (or any Notification Service that the IPP Printer might be configured to use) supplies and localizes the text value of the "human-readable-report" attribute in the Notification according to the charset and natural language requested in the notification subscription.

9 Security Considerations

The IPP Model and Semantics document [ipp-mod] discusses high level security requirements (Client Authentication, Server Authentication and Operation Privacy). Client Authentication is the mechanism by which the client proves its identity to the server in a secure manner. Server Authentication is the mechanism by which the server proves its identity to the client in a secure manner. Operation Privacy is defined as a mechanism for protecting operations from eavesdropping.

~~If we add the 'successful-ok-but-cancel-subscription' (see ISSUE 3 in section 3), then~~ The Notification Recipient can cancel unwanted Subscriptions created by other parties without having to be the owner of the subscription by returning the 'successful-ok-but-cancel-subscription' status code in the Send-Notifications response returned to the Notification Source.

9.1 Security Conformance

Notification Sources (client) MAY support ~~:~~Digest Authentication [rfc2617]. If Digest Authentication is supported, then MD5 and MD5-sess MUST be supported, but the Message Integrity feature NEED NOT be supported.

- ~~Digest Authentication [rfc2069]~~.

~~-MD5 and MD5-sess MUST be implemented and supported.~~

- ~~The Message Integrity feature NEED NOT be used~~.

Notification Recipient (server) MAY support Digest Authentication [rfc2617]. If Digest Authentication is supported, then MD5 and MD5-sess MUST be supported, but the Message Integrity feature NEED NOT be supported.~~:~~

~~-Digest Authentication [rfc2069].~~

~~-MD5 and MD5-sess MUST be implemented and supported.~~

~~-The Message Integrity feature NEED NOT be used.~~

Notification Recipients MAY support TLS for client authentication, server authentication and operation privacy. If a notification recipient supports TLS, it MUST support the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite as mandated by RFC 2246 [rfc2246]. All other cipher suites are OPTIONAL. Notification recipients MAY support Basic Authentication (described in HTTP/1.1 [rfc2616~~rfc2068~~]) for client authentication if the channel is secure. TLS with the above mandated cipher suite can provide such a secure channel.

10 References

[ipp-mod]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", <draft-ietf-ipp-model-v11-04.txt>, June, 1999.

[ipp-ntfy]
Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R., "Internet Printing Protocol/1.1: IPP Event Notification Specification", <draft-ietf-ipp-not-spec-01.txt>, October 14, 1999.

[ipp-pro]
Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.1: Encoding and Transport", draft-ietf-ipp-protocol-v11-03.txt, June, 1999.

[rfc2026]
S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

~~[rfc2068]~~
~~R. Fielding, et al, "Hypertext Transfer Protocol – HTTP/1.1" RFC 2068, January 1997.~~

~~[rfc2566]~~
~~deBry, R., Hastings, T., Herriot, R., Isaacson, S., Powell, P., "Internet Printing Protocol/1.0: Model and Semantics", RFC 2566, April 1999.~~

[rfc2616]
R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

[rfc2617]
J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

11 Author's Addresses

Hugo Parra
Novell, Inc.
122 E 1700 S
Provo, UT 84606

Phone: 801-861-3307
Fax: 801-861-2517
e-mail: hparra@novell.com

Tom Hastings
Xerox Corporation

737 Hawaii St. ESAE 231
El Segundo, CA 90245

Phone: 310-333-6413
Fax: 310-333-5514
e-mail: hastings@cp10.es.xerox.com

12 Full Copyright Statement