Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 29, 2020

R. Van Rein
InternetWide.org

January 26, 2020

User Names for HTTP Resources
draft-vanrein-http-unauth-user-03

Abstract

Most protocols support users under domain names, but HTTP does not. Usage patterns in the wild do suggest a desire to have this facility. This specification defines a header for user names, orthogonal to any authentication or authorisation concerns.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 29, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Most protocols support Network Access Identifiers [RFC7542] like john@example.com to identify users like john under domains such as example.com. The URI format for HTTP can express [Section 2.7.1 of [RFC7230]] such authority sections, and many online applications seem to want to address individual users, but HTTP URIs do not usually express user names. This specification therefore introduces a header "User", in close parallel to the "Host" header.

Historically, user names have been coupled to (Basic and Digest) authentication. This is not generally correct; the user name in the URI specifies a resource name space, not an (authenticated) client identity. By using a new header field, this specification allows authentication to be orthogonal to resource name space selection.

Some user agents have supported (Basic and Digest) authentication with a "user:password" format in the authority section of URIs. This has now been deprecated [Section 3.2.1 of [RFC3986]] but the form with just "user" and no ":password" continues to be acceptable. Various HTTP clients have different handling for this form, sometimes flagging it incorrectly as a security hazard, which also motivates a specification for proper handling.

The purpose of this specification is to define clear meaning for HTTP URIs with a user name.

2. The HTTP User Header

The "User" header field provides an aspect of the desired resource name scope. The value is usually taken from the authority section [Section 3.2 of [RFC3986]] of the target URI and MUST NOT include a ":" colon (U+003a) character.

The User header value holds precisely one value with the following ABNF grammar:

```
User = 1*( unreserved / pct-encoded / sub-delims )
```

The referenced non-terminals are as for URIs [RFC3986] and can be directly included in the quoted-string form; a plain token cannot express "(", ")", "=", ";" and "," without escaping [Section 3.2.6 of [RFC7230]].

3. Protocol Handling of HTTP User

User agents SHOULD render user names in authority sections whenever they render host names, though it may be helpful if it stands out graphically [Section 7.6 of [RFC3986]]. User agents SHOULD NOT remove user names from the target URI. User agents MAY remove the "@" (U+0040) symbol from a URI when the preceding user name is empty.

User agents MUST reject userinfo sections containing a colon ":" (U+003a) or URI syntax errors and MAY warn about potential security problems when they contain a dot "." (U+002e), but SHOULD accept and pass all other non-empty userinfo sections that conform to URI syntax in a User header.

The User header MAY appear in requests and MUST NOT occur in responses.

When sending it, the user agent SHOULD generate User as the next header field after Host. Transparent intermediates such as proxies and caches MUST NOT add, remove or modify the User header. The CONNECT method and Host header both exclude this information, and the User header completes it.

Servers MAY ignore the User header [Section 3.2.1 of [RFC7230]]. When they use it, the Effective Request URI [Section 5.5 of [RFC7230]] is constructed with the userinfo and the at "@" delimiter (U+0040) prefixed to the host name and optional port. Although authentication is orthogonal to resource selection, the scope of a realm is scoped under the authority section [Section 2.2 of [RFC7235]] and so the userinfo partitions realms.

HTTP caches [RFC7234] derive no privacy or security concerns from the User header, but they do need to to differentiate requests based on it. To accommodate that, the Vary header [Section 7.1.4 of [RFC7231]] MUST be generated by the server in the matching response, and the header MUST either be a single "*" star (U+002a) or list the "user" name, for all responses whose processing was influenced by the User header. This requirement has no bearing on server software and configurations that ignore the User header.

During redirects or other traversals to (relative) HTTP URIs, the user name MUST be overwritten when the new URI specifies an authority component, and it MUST be kept otherwise. User agents MUST refuse URIs with non-empty userinfo sub-component that do not conform to the User header grammar; user agents MUST send any other non-empty userinfo sub-components as the value of the User header in requests for the target URI.

4. Orthogonality of Authentication (Example)

The user name in a URI refines the resource selection process on a host, but it is easily confused with the orthogonal concept of authentication. Below is an example to demonstrate how these concepts relate intuitively, but only as the result of access control, which is a local choice on the server but not a specification-driven connection. By demonstrating group access, the example shows a less restrictive model that derives from this orthogonality of concepts.

The remainder of this section is informative.

John and Mary both work at the Sales department of Example, Inc. John has written a document and wants Mary to review it. Mary opens a link to the document name space under the sales account at https://sales@example.com/docs and her user agent sends:

```
GET /docs HTTP/1.1
Host: example.com
User: sales
```

The server redirects to add a slash, and when this is specific to the sales account, it must inform caches about this with the Vary header:

```
HTTP/1.1 301 Moved Permanently
Location: /docs/
Vary: User
```

Since the new location lacks an authority component, this part is retained from the referring URI, and the user agent redirects to https://sales@example.com/docs/ and sends:

```
GET /docs/ HTTP/1.1
Host: example.com
User: sales
```

By this time, the server runs into access control, and decides that it needs an authenticated client identity. To this end, it responds with a challenge to the Documents realm:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Knock realm="Documents"
Vary: User
```

Mary's user agent needs to collect credentials, and may hint at the user name "sales" from the URI but, this being the name of a shared resource, Mary has no credentials and instead authenticates as "mary":

```
GET /docs/ HTTP/1.1
Host: example.com
User: sales
Authorization: Knock realm="Documents", user="mary", ...
```

At some point, the server accepts Mary's authentication and proceeds to access control. This phase checks if user "mary" may access realm "Documents" of "https://sales@example.com" by checking that Mary works for the Sales department. Once this is assured, the server returns the requested document list:

```
HTTP/1.1 200 OK
Vary: User
Content-Type: text/html

...
<a href="/docs/review.cgi?docid=123">Review 123 now</a>
...
```

Mary clicks on the link to /docs/review.cgi?docid=123 and her user agent sees a relative reference with no authority component, so this is again used from the referring URI. The new URI therefore becomes https://sales@example.com/docs/review.cgi?docid=123 for which the user agent sends:

```
GET /docs/review.cgi?docid=123 HTTP/1.1
Host: example.com
User: sales
Authorization: Knock realm="Documents", user="mary", ...
```

After access control, the server starts the CGI script with environment variables LOCAL_USER=sales and REMOTE_USER=mary of which only the latter is an authenticated result. The script interprets the LOCAL_USER as a group account and the REMOTE_USER as the acting group member, and returns a page for review of the document and Mary can get to work.

5. IANA Considerations

IANA adds the following entry to the Message Headers registry:

Header Field Name	Template	Protocol	Status	Reference
User		http	TBD	TBD:THIS_SPEC

6. Security Considerations

The User header field as defined herein is orthogonal to issues of authentication or authorisation, and adds no security concerns.

7. Normative References

[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, <https://www.rfc-editor.org/info/rfc3986>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <https://www.rfc-editor.org/info/rfc7234>.

[RFC7235] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Authentication", RFC 7235, DOI 10.17487/RFC7235, June 2014, <https://www.rfc-editor.org/info/rfc7235>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/info/rfc7542>.

Appendix A. HTTP User Environment Variable

The following variable SHOULD be passed up to applications that run on top of the HTTP stack in a server:

LOCAL_USER gives the HTTP User header value after grammar checking and percent-decoding. Like the customary variables HTTP_HOST and PATH_INFO, this specifies the resource being requested. The HTTP_USER header does not describe the identity of the HTTP client, which usually lands in REMOTE_USER after authentication.

Author's Address

Rick van Rein
InternetWide.org
Haarlebrink 5
Enschede, Overijssel 7544 WP
The Netherlands

Email: rick@openfortress.nl