PRECIS
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2014

Y. Oiwa
RISEC, AIST
T. Nemoto
Keio University
B. Kihara
Lepidum

July 8, 2013

HTTPAuthPrep: PRECIS profile for HTTP Authentication
draft-oiwa-precis-httpauthprep-00

Abstract

This document describes how to handle Unicode strings representing user names and passwords for HTTP authentication.

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 http://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 January 9, 2014.

Copyright Notice

Table of Contents

1. Introduction

1.1. Overview

This document describes how to handle Unicode strings representing user names and passwords for HTTP authentication.

For a long time starting HTTP/1.0 [RFC1945], character encodings of HTTP authentication related parameters are defined and handled quite loosely. [RFC1945] defined user-names of Basic authentication to be a subset of ASCII strings (token), and passwords to be assumed as ISO-8859-1 by recipients. Initial version of HTTP/1.1 [RFC2068] and later revisions define grammar rules which indirectly (through the definition of "TEXT" element) insist that both user-names and passwords are in ISO-8859-1. In any way, these definitions are quite often disregarded, and implementations tend to use their local character sets and encodings, which has caused several interoperability problems.

At the time of being (writing this document), the most promising way of solving this problem is to use Unicode [UNICODE] character set along with UTF-8 encoding [RFC3629] as a common vehicle. However, just using UTF-8 does not completely solve the problem, or even makes it worse, because of the non-unique encoding nature of Unicode character sets.

Recently, a PRECIS [I-D.ietf-precis-framework] framework is being standardized to cover this problem set. It defines a framework to resolve non-uniqueness problem of Unicode character sets for information-comparison purposes, especially useful for user identifications. This document describes how to apply the PRECIS framework for general HTTP user authentications, who to implement such framework, and how to use it.

1.2. Applicability

The rules defined in this document can be used in two ways: one way is to use them as MUST- (or SHOULD-) obey rules, by referring it from another standard or non-standard document. In such case, the rules defined in this document will have a normative property.

Another way is to use them as "best current practices", when some specific HTTP authentication scheme does not define any specific method of string preparations. In such case, any implementations are not required to implement (or not to implement) the string preparation rule in this document, but using it may sometimes improve interoperability between implementations.

Any specific authentication scheme MAY define its own string preparation method, especially when an underlying software layer supporting the authentication scheme (such as SASL) defines (or recommends) its own string preparation method. In such cases, implementations SHOULD NOT use the preparation rules described in this document, and these SHOULD obey the scheme-specific requirement.

It is not feasible to implement the string preparation within all HTTP implementations. For interoperability of authentication process, only a small portion of involved softwares are required to actually implement the string preparation algorithms. To this purpose, general application notes are provided in the latter part of this document.

1.3. Terminology

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].

2. Rules

This section defines two PRECIS string preparation rules for user names and passwords.

Note: the RFC 2119 requirements keywords such as "MUST" within this section are effective as the RFC keywords only when the application of these rules are either REQUIRED or RECOMMENDED by any authentication scheme definitions.

2.1. User Names

2.1.1. Definition

User names are strings to identify the user in HTTP authentication.

```
username       = 1*(idpoint)
                 ;
                 ; an "idpoint" is a UTF-8 encoded
                 ; Unicode code point that conforms to
                 ; the PRECIS "IdentifierClass"
                 ;
```

Note that some authentication schemes like Basic MAY restrict several characters to be used in username.

Note also that some authentication schemes like Digest modifies users' inputs to other forms like quoted-string. This document specifies only string preparation.

2.1.2. Preparation

A user name MUST NOT be zero bytes in length. This rule is to be enforced after any normalization and mapping of code points.

Each username MUST conform to the definition of the PRECIS IdentifierClass provided in [I-D.ietf-precis-framework], where the width mapping, additional mapping, case mapping, normalization, and directionality rules are as described below.

1. Fullwidth and halfwidth characters MUST be mapped to their decomposition equivalents.
2. Additional mappings SHOULD NOT be applied, such as those defined in [I-D.ietf-precis-mappings], unless there are implementation-dependent reasons to do so, or these are exceptionally required by specific authentication schemes.
3. Case mapping is not applied.
4. Unicode Normalization Form C (NFC) MUST be applied to all characters.

With regard to directionality, the "Bidi Rule" provided in [RFC5893] applies.

2.2. Passwords

2.2.1. Definition

Passwords are strings to authenticate the user in HTTP authentication.

```
   password       = 1*(freepoint)
                    ;
                    ; a "freepoint" is a UTF-8 encoded
                    ; Unicode code point that conforms to
                    ; the PRECIS "FreeformClass"
                    ;
```

Note that some authentication schemes MAY restrict several characters to be used in passwords.

2.2.2. Preparation

A password MUST NOT be zero bytes in length. This rule is to be enforced after any normalization and mapping of code points.

A password MUST be treated as follows, where the operations specified MUST be completed in the order shown:

1. Width mapping is not applied.
2. Map any instances of non-ASCII space to ASCII space (U+0020).
3. Case mapping is not applied.
4. Apply Unicode Normalization Form C (NFC) to all characters.
5. Ensure that the resulting string conforms to the definition of the PRECIS FreeformClass.

With regard to directionality, the "Bidi Rule" (defined in [RFC5893]) and similar rules do not apply.

3. Application Notes

Implementation of the above rules are sometimes resource-consuming and not realistic, especially when the implementation is not aware of any Unicode string and is handling the authentication credentials as opaque byte strings. This section provides a general application notes for how to realize the above string preparation in the real software.

Note: the note for RFC keywords in the previous section does apply also for this section.

The general principle for the application is: "to send the string correctly, by some means." In particular, if there is "some" provision (either manually or automatically) to ensure the correct encoding and preparation of string at the time of sending, it is considered enough. As a definitive rule, the following provisions are to be taken:

- Recipient side (i.e. HTTP servers) MAY omit any part of string preparation, including Unicode normalization. It MAY process any received strings as is.
- Senders which forward already-prepared strings (i.e. HTTP proxies etc.) MAY omit any part of string preparation.
- Interactive clients which receive human users' input, as form of "characters", have an obligation to prepare the input string into a correct UTF-8 string with regards to the scheme-specific preparation rules. When the authentication scheme specifies that the preparation is a MUST, they MUST do it.
- Clients which receive credentials in a form of "list of octets" (such as those within configuration files) MAY require its users to prepare the string correctly within configuration phases, and MAY omit any part of string preparation at runtime.

As a reverse to these rules, any recipients MUST be prepared to receive any unprepared byte lists or character lists as inputs. Such recipients MAY prepare the string by its own, MAY reject such inputs explicitly, or MAY process these inputs silently when it will lead to failed authentication attempts. However, Such recipients MUST NOT process such inputs in any way which leads to false authentication successes (modulo cryptographically negligible level of probabilities).

4. Design principles

Note: the content of this section is not normative.

The design of the rules provided in previous sections are made under the following concerns and observations.

- ASCII transparency:
  Every code-point within U+0020 to U+007E MUST be preserved, distinguished, and mapped to its one-byte equivalent respectively. This is a strong requirement for compatibility with existing HTTP authentication.
- Latin-1 preservation:
  Code-points U+00A1 to U+00FE SHOULD be preserved and distinguished in the output string. This enables non-crashing mapping from existing ISO-8859-1 user databases, and, if applicable, enables backward-compatible server-side implementation for Basic plain-text authentication.
- Case (non-)mapping:
  As a subset of the above two rules, no case mapping shall be applied (as a basic rule). Without

this, some existing user database will become non-useful, especially when it has already used a non-mapped credentials, and its entries are hashed or one-way encrypted. The opposite case (case-mapped existing databases) can be at least worked around by users, server implementations, or both.

Of course, some authentication schemes designed for specific use-cases can always define a case-folded mappings whenever needed.

- Strong normalizations:
  All representations (decomposed and composed) of every single "character" within Unicode MUST be normalized to exactly one UTF-8 byte sequence. Without this, virtually all "non-Basic" authentication may become broken with regard to internationalized username and passwords (e.g. Digest).

5. Security Considerations

As mentioned previously, any recipients MUST NOT assume that senders will always send a correctly-prepared strings. Care must be taken that incorrectly-prepared strings MUST lead to either a correct result or an authentication failure.

6. IANA Considerations

[[TBD: more precise IANA Considerations here.]]

The IANA shall add the following two entries to the PRECIS Usage Registry:

Applicability: User Names in HTTP Authentication.
Base Class: IdentifierClass.
Subclass: No.
Replaces: No.
Width Mapping: Map fullwidth and halfwidth characters to their decomposition equivalents.
Additional Mappings: None.
Case Mapping: None.
Normalization: NFC.
Directionality: The "Bidi Rule" defined in RFC 5893 applies.
Specification: RFC XXXX. [Note to RFC Editor: please change XXXX to the number issued for this specification.]

Applicability: Passwords of HTTP Authentication.
Base Class: FreeformClass
Subclass: No.
Replaces: No.
Width Mapping: None.
Additional Mappings: Map non-ASCII space to ASCII space (U+0020).
Case Mapping: None.
Normalization: NFC.
Directionality: The "Bidi Rule" defined in RFC 5893 does not apply.
Specification: RFC XXXX. [Note to RFC Editor: please change XXXX to the number issued for this specification.]
---

7. References

7.1. Normative References

[I-D.ietf-precis-framework]	Saint-Andre, P. and M. Blanchet, "PRECIS Framework: Preparation and Comparison of Internationalized Strings in Application Protocols," draft-ietf-precis-framework-08 (work in progress), April 2013 (TXT).
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC3629]	Yergeau, F., "UTF-8, a transformation format of ISO 10646," STD 63, RFC 3629, November 2003 (TXT).
[RFC5893]	Alvestrand, H. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)," RFC 5893, August 2010 (TXT).
[UNICODE]	The Unicode Consortium, "The Unicode Standard, Version 6.1," 2012.

7.2. Informative References

[RFC1945]	Berners-Lee, T., Fielding, R., and H. Nielsen, "Hypertext Transfer Protocol -- HTTP/1.0," RFC 1945, May 1996 (TXT).
[RFC2068]	Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC 2068, January 1997 (TXT).
[I-D.ietf-precis-mappings]	Yoneya, Y. and T. NEMOTO, "Mapping characters for PRECIS classes," draft-ietf-precis-mappings-02 (work in progress), May 2013 (TXT).

Appendix A. Document History (to be removed)

Initial submit.

Authors' Addresses

Yutaka Oiwa
National Institute of Advanced Industrial Science and Technology
Research Institute for Secure Systems
3-11-46 Nakouji
Amagasaki, Hyogo
JP
Email: mutual-auth-contact-ml@aist.go.jp

Takahiro Nemoto

Keio University
Graduate School of Media Design
4-1-1 Hiyoshi, Kohoku-ku
Yokohama, Kanagawa 223-8526
Japan
Email: t.nemo10@kmd.keio.ac.jp

Boku Kihara
Lepidum Co. Ltd.
#602, Village Sasazuka 3
1-30-3 Sasazuka
Shibuya-ku, Tokyo
Japan