(Background of) PRECIS HTTPauthprep proposal

大岩　寛 (ÔIWA, Yutaka)

PRECIS WG, IETF 87 Berlin

HTTPauthprep, technically

- From PRECIS technical point of view:
  - One of the simplest profile so far
    - No special mapping to be applied
    - No case mapping
    - NFC
    - That's (almost) all
- Why needed?
- What needed to be documented?

HTTP auth and I18N (1)

- Broken for a long time
  - No direct mention about text encoding
  - ISO-8859-1 *indirectly* implied
    - ABNF Gramartical rules changed time to time
    - But all of these refer to TEXT (*TEXT)
    - They say, in another context, that

    "*TEXT" in headers are ISO-8859-1 if beyond ASCII
  - Implementations has broken for a long time
    - Old IE, Netscape and others refer local codepage
    - Recent browsers tend to use ISO-8859-1

HTTP auth and I18N (2)

- The time to fix
  - HTTPAUTH WG
    - will define UTF-8 charset for Basic and Digest
      - But how about normalization and preparation?
    - will define experimental new auth scheme
      - New ones will be able to implement correct I18N

HTTP auth and I18N (3)

- Hard to apply
  - Wide varieties of implementations/use cases
    - Web browsers
      - interactive use
      - XMLHTTP client requests – a kind of automatic use
    - Service-specialized client applications
    - Performance-tuned servers
    - Single-feature command-line clients
    - Micro implementations such as M2M

HTTP auth and I18N (3)

- Hard to apply
  - Several conflicting natures
    - Browsers SHOULD (or MUST) do some kind of string preparation
      - Otherwise, I18N will be broken
    - For most servers and simple clients, user-names etc. are just binary blobs
    - M2M environments, with few tens of kilobytes of memory available
      - Even UTF-8 handling is unrealistic
  - We have to find a good meet point

HTTP auth and I18N (4)

- What we need is:
  - A PRECIS-based way of HTTP auth I18N, which:
    - Can solve I18N issue on browser-based user authentication
    - Can be used as a strong (MUST/SHOULD) requirements for future authentication schemes
    - Can also be used as a loose guidance for use with existing authentication schemes
    - Can be “applied” on simple-use HTTP clients without even implementing UTF-8 conversions
      - Especially, ASCII-only clients must be compliant as is

Draft HTTPauthprep

- Actually, a mixture of rule definition and good practice guidance
  - PRECIS rules (section 2, normative (if wanted))
    - Normative only when someone says "you MUST apply HTTPauthprep rules"
  - Usage guidance (section 3, mostly informative)
    - Security consideration (section 5)
  - Design principle (section 4, informative)

Draft HTTPauthprep

- Applicability
  - Only a “default” preparation for HTTPauth
    - Intended SHOULD requirement for new schemes
    - BCP for existing cases without any specifications
  - Specific auth schemes should use specific rules
    - For example, SASL-bound auth schemes SHOULD use SASLprep(bis), not HTTPauthprep

Technical details

- Natures of HTTP authentication (1)
  - User names: will be used both as an ID lookup key and as a *hash input*
    - It is common to use "H(user, pass, others)" as an actual credential
  - Passwords: will be used either as a direct comparable string or as a *hash input*
  - That is, both of them must be "*binary-agreed*" between servers and clients

Technical details

- Natures of HTTP authentication (2)
  - Backward compatibility with existing schemes and existing systems
    - Current rules are “ASCII-only”, “binary-comparable”
    - When needed, case mapping on server side, only with Basic scheme
      - Digest will not work with case mapping
      - There is no standard (CAPITAL/lower) for mapping
  - It must be ASCII transparent
  - It must be useful with both existing case-sensitive/case-insensitive DBs

Outcome

- Case mapping: SHOULD NOT
  - MUST NOT (not even OPTIONAL)
    for general-purpose clients
    (both interactive and command-line)
    - OPTIONAL mapping will break interoperability
  - Application-specific clients MAY use their own rule (either CAPITAL/lower) on “client-side”
    - No default to lower case
- PRECIS process on client side only
  - Server-side mapping will break authentication process based on cryptography and hashes

Comparisons

		HTTPAuthprep		SASLprepbis		Nickname	XMPPbis	
		User name	Password	User name	Password		Local	Resource
Precis class		ID	Free	ID	Free	Free	IDsub	Free
Width mapping		○	-	○	-	-	○	○
Additional mapping	Delimiter mapping	-	-	MAY	-	-	○	○
	Spacial mapping	-	○	MAY	○	○	-	-
	Local case mapping	-	-	MAY	-	-	-	-
Case mapping		-	-	MAY	-	○	○	○
Normalization	NFC	○	○	○	○		○	○
	NFKC					○		
Bidi rule		○	○	○	○	○	○	○
Others	Removing leading/trailing white spaces	-	-	-	-	○	-	-
	Mapping space sequence to one space	-	-	-	-	○	-	-

(Columns exc. HTTPAuthprep: thanks to Nemoto-san)

Usage guideline

- How to apply
  - Not: “software SHOULD implement”
  - But:
    - “senders SHOULD send prepared string”
    - “recipients MAY omit processing and MAY process received strings as is”
  - For precise texts, see Section 4

Usage guideline

- How to apply
  - “senders SHOULD send prepared string”
    “recipients MAY process as is”
  - Consequences:
    - Browsers (senders) are to be implementing
    - Servers will not do anything, nor proxies
    - Simple clients (handling credentials as binary blob) exempted from doing anything
      - Ask users to provide processed UTF-8 binary blob
      - ASCII-only clients will do nothing

Next steps

- For normative rules (Section 2):
  - We may be good to have “SIMPLEprep”, but:
    - it may become a default one even outside HTTP
  - Can we unify rules to “SHOULD NOT map?”
    - If not, it seems better to be separate ones, as these have a separate requirements
- For semi-informative rules (Sections 3-5):
  - Where should these texts go?
    - If to go separately, the current form is OK
    - If to be unified, we may need an informational docs