

Kerberos Working Group

Internationalization for Extensions

Jeffrey Altman

Guiding Principles

- Use Unicode restricted as needed to provide interoperability and future extensibility
- A single set of string preparation rules for all normalized strings
- Do not restrict Kerberos strings to the subset allowed by IDNA
- Must support IDN components and IDN-based realms

ASN.1

□ RFC1510 / Clarifications:

```
KerberosString ::=  
    GeneralString (IA5String)
```

□ Extensions:

```
KerberosString ::= CHOICE {  
    ia5      GeneralString (IA5String),  
    utf8    UTF8String,  
    ...  
}
```

String Preparation

- SASLprep for all Kerberos strings
- IDN derived principals and realms to be Nameprep prepared
- Open Issue: Full Stop mapping
 - Issues are same for SASL and Kerberos
 - Suggest deferral to SASL

Types of KerberosString usages

- storage

- normalized
- no unassigned code points

- query

- normalized
- unassigned code points allowed

- display

- unnormalized

More ASN.1

Define types derived from KerberosString to convey the appropriate usage:

- KerberosQueryString
- KerberosDisplayString
- KerberosStorageString
- KerberosPasswordString
- KerberosSaltString
- KerberosPrincipalQueryString
- etc

Migration Strategy

- RFC1510/Clarifications messages must use GeneralString; Extensions messages must use UTF-8. This is enforced by ASN.1 constraints.
- Interoperability between mixed environments is ensured when KerberosStrings are restricted to IA5String.
- Extensions must be used when i18n names are used throughout.

Error Conditions

- Extensions KDC / RFC1510 Client:
KDC sends KDC_ERR_ETYPE_NOSUPP if salt cannot be represented as IA5String

- RFC1510 Service / Extensions Client:
If client principal cannot be represented as IA5String, a new error KDC_ERR_SERVICE_TOO_OLD is returned.

Error Conditions

- Cross-realm in mixed environments:
There may be a need to downgrade transited-realm field to RFC1510 string forms. If the contents cannot be represented as IA5String, a new error `KRB5_ERR_TRANSITED_REALM_TRANSLATION` is sent.

RFC3490 (IDNA) vs Kerberos

- IDNA uses Nameprep for string preparation
 - performs case folding to lowercase
 - performs full stop mapping

- $\text{SASLprep}(\text{Nameprep}(\text{Unicode})) \neq \text{SASLprep}(\text{Unicode})$
 - domain-style realms
 - host/* principals with IDN components

- Solution: Apply Nameprep before SASLprep in contexts where we know

Constructing Host-Based Service Principals

- Separate IDN component into labels at full stop (any form)

- For each label:
 - If label begins with ACE prefix, remove prefix: convert to Unicode
 - If label contains one or more 181n characters: apply Nameprep
 - If label contains only IA5String: convert to lowercase

- Join labels by U+002E (full stop)

- Pass string to SASLprep

Constructing Domain-Style Realms

- Realm name without ':' or '/'
- Separated by U+002E (full stop)
- Same as principal construction EXCEPT
lowercasing all-ASCII labels is an open issue

Implications for KDB - kadmin

- KDB should store unnormalized strings for use in UI display, error messages, and compatibility with just-send-8 RFC1510
- Kadmin client must send unnormalized to server
- Client must first apply string preparation to ensure no unassigned code points are used
- Server must apply string preparation and reject strings which contain unassigned code points

Things to Do

- Verify compatibility with just-send-8
- Assign new error messages
- X.500 realm names
- Others?