Network Working Group
Internet-Draft
Updates: 7292,8018 (if approved)
Intended status: Informational
Expires: November 4, 2017

N. Mavrogiannopoulos
Red Hat

May 3, 2017

Internationalized passwords in Password-Based Cryptography Specification
draft-mavrogiannopoulos-pkcs5-passwords-00

Abstract

This memo clarifies the requirements of using internationalized strings as passwords in Password-Based Cryptography Specification version 2.1 [RFC8018] (PKCS#5) and Personal Information Exchange Syntax [RFC7292] (PKCS#12).

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 November 4, 2017.

Copyright Notice

Copyright (c) 2017 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 (http://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.

1. Introduction

Utilizing Internationalized passwords is not known to lead to a consistent user experience. US-ASCII passwords as usually preferred since they are unambiguously interpreted by applications, even though UTF-8 [RFC3629] updates US-ASCII in a backwards compatible way.

The reason for preferring US-ASCII passwords, is the fact that UTF-8 does not imply that strings conforming to it, are unambiguously unique. There are can be various forms of the same string which may look identical to an observer, even though it is being represented by a different byte string. The following are certain issues with using passwords in UTF-8.

- There exist various normalization forms, which result to different data for the same input.
- There is no consistent input form in diverse systems.
- There are various deprecated alphabets which should not be allowed for future compatibility.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Passwords in PKCS#5

The existing PKCS#5 [RFC8018] methods (PBES1, PBES2, PBMAC1) treat passwords as an opaque string and describe the usage of ASCII and UTF-8 strings as a possibility of encoding them. In the interest of interoperability, applications conforming to this specification should encode passwords in UTF-8 NFC form and SHOULD be adhering to the OpaqueString profile (section 4.2 of [RFC7613]).

As an exception to the OpaqueString profile, empty (zero-length) passwords MAY be used, when they are not they result of the [RFC7613] processing. That is, an empty string generated from any non-empty input MUST NOT be used.

4. Passwords in PKCS#12

The PKCS#12 document [RFC7292] defines the use of BMPString passwords (a subset of UTF-16), for its defined encryption methods. This document does not add any further restrictions to the input passwords

of these methods, however it is RECOMMENDED to use of (big-endian) UTF-16 NFC form [NFC] for encoding the password.

Furthermore, when the PKCS#12 container files are combined with methods from PKCS#5 [RFC8018], e.g., AES-CBC-Pad, the passwords SHOULD be adhering to the recommendations in Section 3. In that case, since typically the passwords of the MacData field and the encrypted data match, applications which restricted the MacData password to BMPString set, SHOULD fail when the input password cannot be expressed in that set.

5. Compatibility notes

Note that software wishing to decrypt files with internationalized passwords MAY prepare to handle password encoding methods not adhering to this document. The following paragraphs document existing practices and known bugs in popular software.

5.1. Attempting the password in NFC

The recommendations in the PKCS#5 document are not sufficient to deduce the UTF-8 input form of internationalized passwords. Implementations receiving an internationalized password may attempt decrypting using the password in UTF-8 NFC form.

5.2. OpenSSL's incorrect password conversion

OpenSSL versions prior to 1.1.0 had a bug which always assumed the input password was in the ISO8859-1 character set regardless of the actual character set used on the system. This occurred because it attempted to convert to UTF-16 for the BMPString merely by alternating each byte from the input string with a zero byte to expand to 16 bits.

As an example, consider a PKCS#12 file for which the password is intended to be the following two characters:

U+0102 LATIN CAPITAL LETTER A WITH BREVE

U+017B LATIN CAPITAL LETTER Z WITH DOT ABOVE

For the purpose of this example, the user is operating in a legacy 8-bit locale using the ISO8859-2 character set. The above two characters are thus provided to the application as the bytes 0xC3 0xAF.

The correct form of that password for PKCS#12 key derivation includes precisely those characters in UTF-16 big-endian form as required for

a BMPString: the bytes 0x01 0x02 0x01 0x7B. This is the correct version which any application supporting the use of files for certificates and keys MUST support.

Historical versions of OpenSSL, as noted, would assume that the input bytes were in the ISO8859-1 character set. So the input bytes 0xC3 0xAF would therefore be interpreted as the two characters:

U+00C3 LATIN CAPITAL LETTER A WITH TILDE

U+00AF MACRON

The BMPString used for key derivation in this case would include the bytes 0x00 0xC3 0x00 0xAF.

An application in a non-ISO8859-1 locale can therefore attempt to decrypt such wrongly-created files by treating the input password as if it is a sequence of bytes in ISO8859-1 rather than the locale character set in which it really was provided. The application can generate the BMPString by converting from ISO8859-1 to big-endian UTF-16, and attempt to decrypt the file by deriving the key using that rendition of the password.

6. Security Considerations

All the considerations in [RFC8018] and [RFC7292] apply.

7. IANA Considerations

None.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC7613] Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", RFC 7613, DOI 10.17487/RFC7613, August 2015, <http://www.rfc-editor.org/info/rfc7613>.

[RFC8018] Moriarty, K., Ed., Kaliski, B., and A. Rusch, "PKCS #5: Password-Based Cryptography Specification Version 2.1", RFC 8018, DOI 10.17487/RFC8018, January 2017, <http://www.rfc-editor.org/info/rfc8018>.

[RFC7292] Moriarty, K., Ed., Nystrom, M., Parkinson, S., Rusch, A., and M. Scott, "PKCS #12: Personal Information Exchange Syntax v1.1", RFC 7292, DOI 10.17487/RFC7292, July 2014, <http://www.rfc-editor.org/info/rfc7292>.

[NFC] Davis, M. and M. Duerst, "Unicode Standard Annex #15: Unicode Normalization Forms r.44", Unicode , February 2016.

8.2. Informative References

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

Appendix A. Acknowledgements

The compatibility notes section is based on David Woodhouse's compatibility notes on certificate best practices.

Author's Address

Nikos Mavrogiannopoulos
Red Hat, Inc.
Brno
Czech Republic

Email: nmav@redhat.com