The IAB notes that the IETF community has recently discovered some possible issues with using certain kinds of characters in identifiers. The discovery was as a consequence of examination of new characters added in Unicode 7.0.0. The cases include domain names, and may include user names and other types of strings. The issues arise primarily when the locale information is not carried with the identifier.

The problem happens when a script contains both a precomposed form of a character and decomposed form of a character and those two forms are not canonically equivalent. The first cases discovered involve the use of a *hamza* above other base characters. In principle it could happen in other ways were the Unicode Technical Committee to find that it needed to encode other kinds of characters similarly. Details about the problem are outlined below, under the "Background" heading. Critically, however, use of the affected characters in global, locale-free identifiers presents a possible interoperability issue and a potential security issue: users could be confused, and attacks such as phishing could be enabled.

Because of those issues, until the IETF works out how to address this potential pitfall for identifiers, the IAB recommends (as a temporary measure) that any character using *hamza* above another base character, in either precomposed or decomposed form, be excluded from any new identifier whenever user language information might not be available. This condition certainly exists in all public domain names and quite probably in all domain names; the principles in RFC 6912 apply. In many (perhaps most) cases it also exists in user names and strings like passwords. The IAB does not extend its recommendation to existing identifiers, but suggests caution in such cases.

On the same precautionary principle, the IAB recommends that the Internationalized Domain Names for Applications (IDNA) Parameters registry (http://www.iana.org/assignments/idna-tables/) not be updated to Unicode 7.0.0 until the IETF has consensus on a solution to this problem.

The IAB expects that the IETF protocol community will continue to investigate ways to address this problem, and we suspect that further investigation will broaden the problem's scope. The IAB also recognizes that excluding characters from use in all identifier contexts presents a disadvantage to those Internet users who wish to use such characters to support normal orthography in their language. Our recommendation is wider than strictly needed for the present case, but will protect against any possible similar case that could come up while the IETF sorts out what to do. We recognize the problem is urgent, and ask the IETF to engineer a solution with all deliberate haste.

Background

Unicode 7.0.0 introduced a character, ARABIC LETTER BEH WITH HAMZA ABOVE (U+08A1), that inspired some investigation. The result was the discovery of the issue discussed here. The issue is not new, and appears to affect all versions of IDNA and any other identifier system that has made assumptions about normalization and character comparison.

What is peculiar about these cases, as distinct from other confusable cases, is that the decomposed and precomposed forms are in the same script and cannot be distinguished visually by users, even in large fonts designed for clarity. It is only by knowing the language that it is possible to detect whether a use of the character is the correct one. The reasoning for this may be peculiar to Arabic script; in the absence of an exhaustive analysis of all scripts — those completely encoded, partially encoded, and so far unencoded — we do not have the necessary background to make a determination. In any case, after considerable analysis, we believe that,

whatever the reasoning, these sorts of cases are problematic for IETF-related uses of Unicode for identifiers.

The character that caused this issue to become plain is ARABIC LETTER BEH WITH HAMZA ABOVE (U+08A1). It looks like this:

U+08A1

In versions of Unicode since at least 3.2 and including 7.0.0, it is possible to write the letter *beh* and a combining character *hamza* to get a sequence, ARABIC LETTER BEH (U+0628) and ARABIC HAMZA ABOVE (U+0654):

U+0628 U+0654

It is not unusual to have multiple ways to generate “the same” character in the same script. For instance, in the Latin script, there is the character a with a diaeresis above it. It can be generated as a stand-alone character LATIN SMALL LETTER A WITH DIAERESIS (U+00E4). It looks like this:

U+00E4

In versions of Unicode since at least 3.2 and including 7.0.0, it is possible to write a-diaeresis by writing the letter a and a combining diaeresis to get a sequence: LATIN SMALL LETTER A (U+0061), and COMBINING DIAERESIS (U+0308):

U+0061 U+0308

What is different between the “a” and “BEH” cases is the way these characters are treated. When one applies Normalization Form C (see UAX #15 at http://www.unicode.org/reports/tr15/) to the series U+0061 U+0308, it produces U+00E4; similarly, when one applies Normalization Form D

to U+00E4, it produces U+0061 U+0308. These normalization rules reflect the Unicode observation that these are two different ways of writing the same character. The same is not true of U+08A1 and U+0628 U+0654, however. These pairs are not treated as being the same character. The reasons for this are discussed in section 9.2 of the Unicode Standard, version 7.0.0, pp362 ff (available at http://www.unicode.org/versions/Unicode7.0.0/ch09.pdf). The Standard says roughly that, because one approach is used to write a glottal stop whereas the other is written as a diacritic, the two characters are not the same thing linguistically. (The details are considerably more subtle than this. We refer the interested reader to the original in order to understand the details.)

This approach is at first glance somewhat unusual in the way Unicode treats character formation. For instance, the diaeresis represents a diacritic (umlaut) on a in German. In Swedish, a-with-diaeresis is a completely different letter from a. Yet under Unicode normalization, U+0061 U+0308 and U+00E4 normalize to the same thing (regardless of how the diaeresis is used). As noted, the Unicode Standard has an extended discussion of why things are different for *hamza*. The IAB does not dispute the reasoning that the Unicode Technical Committee uses for such determinations (neither is the IAB qualified to dispute that reasoning). The IAB's sole concern in this case is whether the correct choice of character will be discernible in the context of identifiers. Because identifiers often appear in contexts that are free of language indications, and because identifiers are frequently so short as to defy textual analysis for language, it may be that the handling of the *hamza* in Unicode presents a problem for usable and safe identifiers. This is the reason the IAB is urging caution now, while the issues are sorted out in the IETF.

In Unicode version 7.0.0, we have identified three characters for which the condition discussed above occurs; but, the principles in the relevant section of the Standard (and some discussions) make it clear it could happen for other characters in the future, and that similar situations may already exist in current and earlier versions of Unicode. When there is additional information about the language of the user available, special processing prior to comparisons might be possible, avoiding this problem. However, these characters in global, locale-free identifiers present a possible interoperability issue, because different users will generate what they take to be the same identifiers and yet the identifiers will not match each other. They also represent a potential security issue, because it may be possible to use these characters to deceive users into thinking that they are using an identifier that they are not actually using.

The IETF has been diligently working to update its standards to be Unicode version-agnostic, but it appears that this is an area where different versions of Unicode might produce different results when handling or processing a string: in Unicode version 7.0.0, there is the potential for an issue around the newly-introduced character, whereas in previous versions of Unicode there is only one way to write the character in question (and U+08A1 is unassigned). More generally, because of the potential to create user confusion and to enable attacks, and because we regard the latter as a serious risk to all users whose normal writing system uses Arabic script, the IAB recommends (as a temporary measure) not using *hamza* above a base character, in either precomposed or decomposed form, in any new identifier when user language information is not available.

Updating the IDNA Parameters registry (http://www.iana.org/assignments/idna-tables/) would require treating U+08A1 as PVALID, and one open question is whether that would be a good thing.

Further detailed discussion of this issue may be found in draft-klensin-idna-5892upd-unicode70-03, available in any up to date Internet-Draft repository.

Please note that this IAB statement was edited on 2015-02-11. The original text published on 2015-01-27 is archived here.