ThreatFactor NSIA - Bug #69

Mixed Mode Evaluators

04/08/2010 11:51 PM - Luke Murphey

Status:	Rejected	**Start date:**	04/08/2010
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

The current scan engine does not allow mixed mode operators. This means that evaluators that use offsets based on the character position cannot be mixed with evaluators that rely on offsets based on byte positions.

For example, a rule that perform a string search for a set of characters cannot be followed by a byte evaluator that tests to determine if the following bytes match an array of bytes.

History

#1 - 04/08/2010 11:53 PM - Luke Murphey

This is due to that fact that the current scan engine does not have a method to convert a byte offset to a character offset. In ASCII, one character always equals one byte. However, Unicode presents a series of complications including:

1. The Unicode format must be detected using a similar algorithm as a web-browser (even if the web-browsers do it incorrectly). Otherwise, this can be used as an evasion technique.
2. Multiple Unicode formats exist
3. Unicode allows variable sizes to be associated with characters. For example, UTF-8 characters can be from 1-4 bytes.
4. UTF-16 may include a 2-byte header that it used to indicate if the bytes are low-order or high order.

Normalizing the input to a single encoding (such as UTF-32) will help keep the input size consistent and thus deals with problem 2. However, this may cause issues when attempting to detect exploits that use character encoding interpretation problems as part of the exploit. Therefore, this should be avoided.

All currently used characters are in the BMP (U+0000 to U+FFFF).

Additional reading:

- http://www.tbray.org/ongoing/When/200x/2003/04/26/UTF
- http://en.wikipedia.org/wiki/Comparison_of_Unicode_encodings
- http://www.ibm.com/developerworks/library/codepages.html
- http://gedcom-parse.sourceforge.net/doc/encoding.html
- http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Character.html

#2 - 04/08/2010 11:54 PM - Luke Murphey

- Status changed from New to Rejected

I don't believe this feature is necessary at the current time. The current detection engine is more than sufficient and the amount of work to build a scan engine that understands Unicode to the required depth is substantial.