Authentication Token Summary

Jeff Disher (Copyright 2013 Open Autonomy Inc.)
Date: 2013-08-22
Version: 1

Introduction:

All interaction between components in the system use the same “progressive authentication key sequence” algorithm.

The token has 2 main components:

1) KEY - the key is the component sent over the wire and both the sender and receiver expect exactly the same key for a given RPC message.

2) STRIDE - the stride is the private component and is stored by both the sender and receiver.

Both of these components are created by an identity and they are exchanged as a “packed token” (“VERSION-KEY-STRIDE” for web applications and “VERSION-KEY-STRIDE-IDENTIFIER” for pre-authorized “native” tokens). VERSION is always “0”. KEY and STRIDE are always **lower-case ASCII strings**.

The key sequence can be considered as a half-duplex communication channel between the sender and receiver (half-duplex in that one party is always the “sender” of the communication). Each of these channels is distinct and one channel being compromised or invalidated has no effect on the other channels managed by either party.

Key progression:

Once a key is used, both sides of the channel generate the next key in the sequence by creating a lower-case hex ASCII string of the bytes generated by a SHA1 hash, like so:

NEW_KEY := sha1(concat(KEY, STRIDE))

Authentication Failure:

If an incorrect key is used in a message or either side of the channel deliberately corrupts the next key (typically done to force a “time out” on the channel), an error is returned and the applications either require a re-authentication flow to be initiated or a new key to be commuted to both sides by a user.