Jarvis - Feature #605
Add imwww interface from Jarvis to monitor

28 Jan 2018 17:11 - Hammel

Status:	Closed	**Start date:**	28 Jan 2018
Priority:	Immediate	**Due date:**	
Assignee:	Hammel	**% Done:**	100%
Category:	Messages	**Estimated time:**	0.00 hour
Target version:	0.5.0		
Severity:	01 - Critical		

Description

This is the interface that makes REST API connections to send messages and retrieve responses.

Input to this class comes in Message objects.

Output from this class is delivered in Message objects.

The Message class may be extended to support JSON parsing of string data.

Associated revisions

Revision 9653bfd1 - 06 Apr 2018 15:58 - Hammel

RM #605: Added Imwww class to handle communications via REST API to monitor.

Revision 5e981432 - 23 Apr 2018 16:50 - Hammel

RM #605: Fixed registration to no longer use SSL.

Revision 79240fd7 - 03 May 2018 19:47 - Hammel

RM #605: Added AES crypto support.
Fixed PUT/GET issues when using HttpURLConnection.setDoOutput().
Added support for retrieving monitor description.

Revision 0bb83f8f - 07 May 2018 14:10 - Hammel

RM #605: Implement support for AES decryption for inbound messages from monitor.

Revision ff25cc2d - 08 May 2018 13:24 - Hammel

RM #605: Implemented encryption on Java side for decryption on the monitor side. Tested with temporary code in Register class that requests a device list. That needs to be removed later though it causes no harm.

History

#1 - 28 Mar 2018 20:28 - Hammel

- Priority changed from High to Immediate

- Target version changed from 0.1.0 - Baby Steps to 0.5.0

- Severity changed from 02 - High to 01 - Critical

Moving up in priority.

#2 - 01 Apr 2018 16:28 - Hammel

- Subject changed from Add messaging interface from Jarvis to PiBox to Add imwww interface from Jarvis to monitor

- Description updated

#3 - 06 Apr 2018 16:04 - Hammel

- Status changed from New to In Progress

- % Done changed from 0 to 30

Implemented and partially tested. Still needs to be tested against imwww REST API, however. Also does not actually accept any specific commands from JarvisCmd, so that needs to be added there too before I can verify the command makes it all the way to the monitor.

#4 - 23 Apr 2018 15:02 - Hammel

Implementing SSL has become problematic, probably because I can't quite grasp all the nuances of getting it right (not to mention the problems of dealing with self-signed certs). But it also now seems like overkill. What I really want is the same process the sensors are using with the monitor:

1. Exchange UUID key at registration: each monitor has a unique UUID and each Jarvis node has unique UUID.
2. Encrypt a message with aes(UUID+IV+message).
3. Send IV plus encrypted message in the clear (http, not https)
4. Decrypt using UUID+IV

The UUID is never sent except in the registration so it's a low-probability of getting intercepted - the user has to enable registration. So while the message and IV are sent in the open, the message itself is encrypted. You can know who is talking but not what is being said. At least not easily.

This will simplify message passing considerably. It's probably hackable, but not easily. And even the hacking could be reduced by using a sneaker-net input of remote end point keys.

I need to implement this on both end points: monitor (imrest) and Jarvis.

#5 - 07 May 2018 16:02 - Hammel

- % Done changed from 30 to 50

This has been implemented in Jarvis for decoding messages that were encrypted on on the NodeJS side. It's been shown to work by having Jarvis ask for /monitor, which returns the monitor descriptor using AES encryption inside a JSON packet that contains an IV and a message.

Since AES 128 is being used the encryption key (Jarvis' UUID) is shorted to 16 bytes on both ends.

This code is committed and pushed.

Now I need to reverse the process by encrypting in Jarvis and decrypting on the NodeJS side. This will be tested using the /device API from Jarvis to monitor.

#6 - 08 May 2018 13:29 - Hammel

- Status changed from In Progress to Closed

- % Done changed from 50 to 100

Code implemented, tested and pushed. I verified it worked by having Jarvis encode a device state change request and sent it to the monitor, which printed out the decoded request string.

Closing issue.