Website Monitoring - Bug #1524

Inputs hang each other up (especially ones that timeout)

10/18/2016 04:57 PM - Luke Murphey

Status:	Closed	**Start date:**	10/19/2016
Priority:	Normal	**Due date:**	
Assignee:	Luke Murphey	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2.0		

Description

To save memory, the input currently uses single instance mode (a single input running all of the inputs).

To fix this, I could:

1. Change input away from single instance mode
    1. This is difficult to do because then I would have to use Splunk's interval which doesn't support time specifiers (like 5m).
    2. Would need to a process to convert these to Splunk's interval and the current interval; UI would likely need to accept and convert to/from Splunk's interval
2. Switch to a multi-threading model

Subtasks:

Task # 1529: Change base class to use RLocks	**Closed**
Task # 1530: Create multiple threads for performing pings	**Closed**
Task # 1531: Clean up threads on shutdown	**Closed**

History

#1 - 10/18/2016 04:57 PM - Luke Murphey

- Subject changed from Inputs hang each other up to Inputs hang each other up (especially ones that timeout)

#2 - 10/18/2016 05:28 PM - Luke Murphey

https://docs.python.org/2/library/threading.html

#3 - 10/18/2016 05:28 PM - Luke Murphey

- https://answers.splunk.com/answers/464902/website-monitoring-app-not-working-as-per-the-sche.html
- https://answers.splunk.com/answers/462699/website-monitoring-is-there-a-limit-on-the-number.html
- https://answers.splunk.com/answers/386292/polling-frequency-seems-to-default-to-10m.html
- https://answers.splunk.com/answers/308170/website-monitoring-why-does-monitoring-seem-slow-a.html

#4 - 10/18/2016 06:00 PM - Luke Murphey

- Target version set to 2.0

#5 - 10/19/2016 06:12 PM - Luke Murphey

Things that need to be changed to support multi-threading:

- ~~**logger():**~~ needs to not allow multiple thread access (is thread safe per http://stackoverflow.com/questions/2973900/is-pythons-logging-module-thread-safe)
- **output_result():** needs to control multiple thread access to output_event()
- **run():** needs to instantiate multiple threads
- **shutdown():** needs to cleanup threads

#6 - 10/19/2016 06:12 PM - Luke Murphey

http://effbot.org/zone/thread-synchronization.htm

#7 - 10/19/2016 06:22 PM - Luke Murphey

Uncontesting access to locks doesn't appear to have much of a performance issue :
http://stackoverflow.com/questions/11966471/python-cost-of-locking-vs-performance-does-multithreading-make-sense

#8 - 10/21/2016 05:47 AM - Luke Murphey

- Status changed from New to In Progress

- Assignee set to Luke Murphey

#9 - 10/22/2016 04:44 PM - Luke Murphey

- Status changed from In Progress to Closed