

## Website Monitoring - Feature #606

### Proxy Support

07/18/2013 05:24 PM - Luke Murphey

<b>Status:</b>	Closed	<b>Start date:</b>	07/18/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Luke Murphey	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	0.8		
<b>Description</b> Add the ability to use a proxy.			

#### Associated revisions

##### Revision 44 - 08/31/2013 08:08 AM - lmurphey

Added ability to use a proxy server. Reference #606.

#### History

##### #1 - 07/19/2013 02:49 AM - Luke Murphey

See:

- <http://stackoverflow.com/questions/11805773/tunneling-http-lib-through-a-proxy>
- <http://stackoverflow.com/questions/12038213/http-lib-and-https-proxies>

##### #2 - 08/29/2013 08:06 AM - Luke Murphey

This is somewhat supported already by Python:

In a Unix or Windows environment, set the `http_proxy`, or `ftp_proxy` environment variables to a URL that identifies the proxy server before starting the Python interpreter.

...

In a Windows environment, if no proxy environment variables are set, proxy settings are obtained from the registry's Internet Settings section.

In a Mac OS X environment, `urlopen()` will retrieve proxy information from the OS X System Configuration Framework, which can be managed with Network System Preferences panel.

[[Source](#)]

##### #3 - 08/30/2013 03:16 AM - Luke Murphey

See [https://github.com/LukeMurphey/splunk-website-monitoring/blob/master/src/bin/web\\_ping.py#L197](https://github.com/LukeMurphey/splunk-website-monitoring/blob/master/src/bin/web_ping.py#L197) for the HTTPS connection invocation.

**#4 - 08/30/2013 04:59 AM - Luke Murphey**

- Description updated

**#5 - 08/30/2013 05:11 AM - Luke Murphey**

See [http://docs.python.org/2/library/httplib.html?highlight=set\\_tunnel#httplib.HTTPConnection.set\\_tunnel](http://docs.python.org/2/library/httplib.html?highlight=set_tunnel#httplib.HTTPConnection.set_tunnel)

**#6 - 08/30/2013 08:20 AM - Luke Murphey**

Problem is that httplib does not return the response code of the website but of the tunnel.

httplib2 has better support: <https://httplib2.googlecode.com/hg/doc/html/libhttplib2.html>

See also: <http://stackoverflow.com/questions/2907812/does-httplib2-support-http-proxy-at-all-socks-proxy-works-but-not-http>

**#7 - 08/30/2013 08:24 AM - Luke Murphey**

- Status changed from New to In Progress

**#8 - 08/30/2013 08:24 AM - Luke Murphey**

- % Done changed from 0 to 50

**#9 - 08/30/2013 09:59 PM - Luke Murphey**

This might not be working with urls that have an alternative port

**#10 - 08/31/2013 06:59 AM - Luke Murphey**

Docs for httplib2 are here: <http://httplib2.googlecode.com/hg/doc/html/index.html>

**#11 - 08/31/2013 08:10 AM - Luke Murphey**

- % Done changed from 50 to 90

**#12 - 09/01/2013 02:49 AM - Luke Murphey**

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

**#13 - 09/04/2013 09:29 PM - Luke Murphey**

Someone is getting an error:

```
2013-09-02 15:29:26,198 ERROR Execution failed: Traceback (most recent call last):
File "/opt/splunk/etc/apps/website_monitoring/bin/modular_input.py", line 844, in execute
self.do_run(in_stream, log_exception_and_continue=True)
File "/opt/splunk/etc/apps/website_monitoring/bin/modular_input.py", line 748, in do_run
input_config)
File "/opt/splunk/etc/apps/website_monitoring/bin/web_ping.py", line 497, in run
proxy_type, proxy_server, proxy_port, proxy_user, proxy_password = self.get_proxy_config(input_config.session_
key)
File "/opt/splunk/etc/apps/website_monitoring/bin/web_ping.py", line 479, in get_proxy_config
website_monitoring_config = WebsiteMonitoringConfig.get( WebsiteMonitoringConfig.build_id( "default", "website
_monitoring", "nobody"), sessionKey=session_key )
File "/opt/splunk/lib/python2.7/site-packages/splunk/models/base.py", line 542, in get
return SplunkRESTManager(cls, sessionKey=sessionKey).get(id)
File "/opt/splunk/lib/python2.7/site-packages/splunk/models/base.py", line 522, in get
entity = self._get_entity(id, host_path=host_path)
File "/opt/splunk/lib/python2.7/site-packages/splunk/models/base.py", line 438, in _get_entity
return self._fix_entity(splunk.entity.getEntity(self.model.resource, None, sessionKey=self.sessionKey, uri=id)
)
File "/opt/splunk/lib/python2.7/site-packages/splunk/entity.py", line 211, in getEntity
serverResponse, serverContent = rest.simpleRequest(uri, getargs=kwargs, sessionKey=sessionKey, raiseAllErrors=
True)
```

```

File "/opt/splunk/lib/python2.7/site-packages/splunk/rest/__init__.py", line 436, in simpleRequest
serverResponse, serverContent = h.request(uri, method, headers=headers, body=payload)
File "/opt/splunk/etc/apps/website_monitoring/bin/httpplib2/__init__.py", line 1570, in request
(response, content) = self._request(conn, authority, uri, request_uri, method, body, headers, redirections, ca
chekey)
File "/opt/splunk/etc/apps/website_monitoring/bin/httpplib2/__init__.py", line 1317, in _request
(response, content) = self._conn_request(conn, request_uri, method, body, headers)
File "/opt/splunk/etc/apps/website_monitoring/bin/httpplib2/__init__.py", line 1252, in _conn_request
conn.connect()
File "/opt/splunk/etc/apps/website_monitoring/bin/httpplib2/__init__.py", line 1018, in connect
sock.connect((self.host, self.port))
File "/opt/splunk/etc/apps/website_monitoring/bin/httpplib2/socks.py", line 424, in connect
self.__negotiatehttp(destpair[0], destpair[1])
File "/opt/splunk/etc/apps/website_monitoring/bin/httpplib2/socks.py", line 390, in __negotiatehttp
raise HTTPError((statuscode, statusline[2]))
HTTPError: (403, 'Forbidden')

```

```

try:
    import httpplib2
    for aTry in range(tries):
        h = httpplib2.Http(timeout=timeout, disable_ssl_certificate_validation=True)
        if getWebKeyFile() and getWebCertFile():
            h.add_certificate(getWebKeyFile(), getWebCertFile(), '')
        serverResponse, serverContent = h.request(uri, method, headers=headers, body=payload)
        if serverResponse == None:
            if aTry < tries:
                time.sleep(wait)
        else:
            break

```

#### #14 - 09/04/2013 09:58 PM - Luke Murphey

Note that http2lib configures itself from the environment ([https://code.google.com/p/httplib2/source/browse/python2/httplib2/\\_\\_init\\_\\_.py](https://code.google.com/p/httplib2/source/browse/python2/httplib2/__init__.py)):

```

def proxy_info_from_environment(method='http'):
    """
    Read proxy info from the environment variables.
    """
    if method not in ['http', 'https']:

```

```
return
```

```
env_var = method + '_proxy'
url = os.environ.get(env_var, os.environ.get(env_var.upper()))
if not url:
    return
pi = proxy_info_from_url(url, method)

no_proxy = os.environ.get('no_proxy', os.environ.get('NO_PROXY', ''))
bypass_hosts = []
if no_proxy:
    bypass_hosts = no_proxy.split(',')
# special case, no_proxy=* means all hosts bypassed
if no_proxy == '*':
    bypass_hosts = AllHosts

pi.bypass_hosts = bypass_hosts
return pi
```

**#15 - 11/03/2013 05:48 AM - Luke Murphey**

- Status changed from Resolved to Closed