RADIUS Authentication - Feature #378

Support Search Head Pooling

05/06/2012 07:25 PM - Anonymous

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Update application to work with search head pooling.

History

#1 - 05/06/2012 07:26 PM - Luke Murphey

- Target version changed from 1.0.0 to 1.0.1

#2 - 05/09/2012 09:00 PM - Luke Murphey

Some sample code to get the base path:

```
import splunk.clilib.bundle_paths as bp
os.path.join(bp.get_base_path(), 'search', 'lookups')
```

#3 - 07/27/2012 11:42 PM - Luke Murphey

- Target version deleted (1.0.1)

#4 - 07/28/2012 12:10 AM - Luke Murphey

- Target version set to 1.1

#5 - 07/31/2012 10:46 PM - Luke Murphey

I'm not sure how or if this can be made search head compatible since the script that currently reads items from the conf system.

Below are some details about how the shared storage is determined:

From **bundle_paths.py**:

```
def get_shared_storage():
    """
    If search head pooling is enabled, return the location of shared storage.
    Otherwise, return an empty string.
    """
    try:
        state   = comm.getConfKeyValue('server', 'pooling', 'state')
        storage = comm.getConfKeyValue('server', 'pooling', 'storage')
        if ((state == 'enabled') and (len(storage) > 0)):
            return storage
    except Exception, e:
        pass
    return ''
```

From **cli_common.py**:

```
def getConfKeyValue(confName, stanza, key):
  stanza = getConfStanza(confName, stanza)
  try:
    return stanza[key]
  except KeyError:
    raise ParsingError, "no '%s' key exists in the '%s' stanza in %s.conf.  Your configuration may be corrupt 
or may require a restart." % (key, stanza, confName)

def getServerConfKeyValue(key):
  return getConfKeyValue("server", "sslConfig", key)
def getWebConfKeyValue(key):
  return getConfKeyValue("web", "settings", key)

def getConfStanza(confName, stanza):
  cacheConfFile(confName)
  try:
    return confSettings[confName][stanza]
  except KeyError:
    raise ParsingError, "no '%s' stanza exists in %s.conf.  Your configuration may be corrupt or may require a
 restart." % (stanza, confName)

def cacheConfFile(confName):
  global confSettings
  if confName in confSettings:
    return # do nothing if cache exists.

  # cli.py sets this value to False as soon as it loads.  the CLI doesn't rely
  # on the cached files because:
  # - it doesn't have to.
  # - the system user running the CLI command may not have permission to write
  #   to the merged config file location.
  # - concurrent CLI invocations would clobber each other's merged files.

  useCachedConfigs = (("useCachedConfigs" in dir(__main__)) and (__main__.useCachedConfigs,) or (True,))[0]
  # this will not always eval to True; see http://www.diveintopython.net/power_of_introspection/and_or.html#d0
e9975, Example 4.18

  mergedPath = os.path.join(splunk_home, "var", "run", "splunk", "merged", confName + ".conf")
  if useCachedConfigs and os.path.exists(mergedPath) and os.path.getsize(mergedPath) > 0:
    logger.debug("Preloading from '%s'." % mergedPath)
    confSettings[confName] = readConfFile(mergedPath)
  else:
    logger.debug("Running btool for '%s.conf'." % confName)
    confSettings[confName] = getMergedConf(confName)
```

#6 - 08/01/2012 06:22 PM - Luke Murphey

- Target version changed from 1.1 to 1.1.1

#7 - 08/13/2012 02:56 AM - Luke Murphey

- Tracker changed from Bug to Feature

- Subject changed from Make App Work with Search Head Pooling to Support Search Head Pooling

#8 - 08/28/2012 03:45 AM - Luke Murphey

- Target version deleted (1.1.1)

#9 - 07/23/2014 09:36 PM - Luke Murphey

- Assignee deleted (Luke Murphey)