Sign in

jpalanco / alienvault-ossim Public

Notifications Fork 93 Star 119

Code Issues 5 Pull requests 1 Actions Projects Wiki Security Insight

alienvault-ossim / os-sim / agent / src / ParserUtil.py

1146 lines (1016 loc) · 46.7 KB

Code Blame Raw

```python
#
# Otherwise you can read it here: http://www.gnu.org/licenses/gpl-2.0.txt
#

#
# GLOBAL IMPORTS
#
import datetime
import json
import os
import pickle
import re
import socket
import time
from hashlib import md5

import GeoIP

GEOIPDB = GeoIP.open("/usr/share/geoip/GeoLiteCity.dat", GeoIP.GEOIP_STANDARD)
#
# LOCAL IMPORTS
#

from SiteProtectorMap import *
from NetScreenMap import *
from Logger import Logger

logger = Logger.logger

#
# GLOBAL VARIABLES
#
DEFAULT_ID = '99999'
DATE_FORMAT_FILE_PATH = '/etc/ossim/agent/plugins/date_config/date_formats.json'
CUSTOM_FORMATS_LOADED = False
HOST_RESOLV_CACHE = {}

PROTO_TABLE = {
    '1': 'icmp',
    '6': 'tcp',
    '17': 'udp',
}

HOST_BLACK_LIST = {}

FIXED_MONTH_TRANSLATE = {
    # ENGLISH
    'jan': 1,
    'feb': 2,
    'mar': 3,
    'apr': 4,
    'may': 5,
    'jun': 6,
    'jul': 7,
    'aug': 8,
    'sep': 9,
    'oct': 10,
    'nov': 11,
    'dec': 12,
    'january': 1,
    'february': 2,
    'march': 3,
    'april': 4,
    # 'May':5,
    'june': 6,
    'july': 7,
    'august': 8,
    'september': 9,
    'october': 10,
    'november': 11,
    'december': 12,
    # SPANISH
    'ene': 1,
    # 'feb':2,
    # 'mar':3,
    'abr': 4,
    # 'may':5,
    # 'jun':6,
    # 'jul':7,
    'ago': 8,
    # 'sep':9,
    # 'oct':10,
    # 'nov':11,
    'dic': 12,
    'enero': 1,
    'febrero': 2,
    'marzo': 3,
    'abril': 4,
    'mayo': 5,
    'junio': 6,
    'julio': 7,
    'agosto': 8
```

```
            'agosto': 8,
```

```
HOST_RESOLV_DYNAMIC_CACHE = {}

# Dynamic host-ip cache.
def refreshCache(data):
    """ Refresh the HOST dynamic cache """
    # action="refresh_asset_list" list={ossim-unstable-pro=192.168.2.18,crosa=192.168.2.130} id
    logger.debug("Updating dynamic host cache... %s" % data)
```

```
        # HostResolv.HOST_RESOLV_DYNAMIC_CACHE.clear()
        pattern = "action=\"refresh_asset_list\"\s+list={(?P<list>.*)}"
        ipv4_reg = "\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"
        hostname_valid = "(([a-zA-Z]|[a-zA-Z][a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z]|[A-Za-z][A-Za-
        reg_comp = re.compile(pattern)
        res = reg_comp.match(data)
        host_list = []
        new_cache = {}
        if res is not None:
            tmp_list = res.group('list')
            if tmp_list is not None:
                host_list = tmp_list.split(';')
                logger.debug("HOST_LIST: %s" % host_list)
                for asset in host_list:
                    if asset == '':
                        continue
                    ip, hostnames = asset.split('=')
                    hostname_list = hostnames.split(',')
                    logger.debug("IP = %s , hostnamelist: %s" % (ip, hostname_list))
                    for hostname in hostname_list:
                        hostname = hostname.strip()
                        hostname = hostname.lower()
                        if re.match(ipv4_reg, ip) and re.match(hostname_valid, hostname):
                            if new_cache.has_key(hostname):
                                if ip not in new_cache[hostname]:
                                    new_cache[hostname].append(ip)
                            else:
                                new_cache[hostname] = []
                                new_cache[hostname].append(ip)

        HostResolv.HOST_RESOLV_DYNAMIC_CACHE = new_cache

        HostResolv.printCache()
        HostResolv.saveHostCache()

    refreshCache = staticmethod(refreshCache)

    def saveHostCache():
        logger.info("Saving dynamic host cache in /etc/ossim/agent/host_cache.dic")
        pickle.dump(HostResolv.HOST_RESOLV_DYNAMIC_CACHE, open("/etc/ossim/agent/host_cache.dic", "

    saveHostCache = staticmethod(saveHostCache)

    def loadHostCache():
        if os.path.isfile("/etc/ossim/agent/host_cache.dic"):
            try:
                logger.debug("Loading dynamic host cache from '/etc/ossim/agent/host_cache.dic'")
                HostResolv.HOST_RESOLV_DYNAMIC_CACHE = pickle.load(open("/etc/ossim/agent/host_cach
                HostResolv.printCache()
            except:
                logger.warning("Deleting corrupt file host_cache_pro.dic")
                os.remove("/etc/ossim/agent/host_cache_pro.dic")
                return False
        else:
            return False

        return True

    loadHostCache = staticmethod(loadHostCache)

    def printCache():
        logger.debug("------------------ Dynamic cache ---------------------")
        for host, ip in HostResolv.HOST_RESOLV_DYNAMIC_CACHE.items():
            logger.debug("%s  -------->> %s" % (host, ip))
        logger.debug("------------------------------------------------------")

    printCache = staticmethod(printCache)
```