Code Explanation Log

File: logsql.py

Norsk forklaring av kode:

```
#*********************************************************************************
#*                            Dionaea
#*                         - catches bugs -
#*
#*
#*
#* Copyright (C) 2009  Paul Baecher & Markus Koetter
#*
#* This program is free software; you can redistribute it and/or
#* modify it under the terms of the GNU General Public License
#* as published by the Free Software Foundation; either version 2
#* of the License, or (at your option) any later version.
#*
#* This program is distributed in the hope that it will be useful,
#* but WITHOUT ANY WARRANTY; without even the implied warranty of
#* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#* GNU General Public License for more details.
#*
#* You should have received a copy of the GNU General Public License
#* along with this program; if not, write to the Free Software
#* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
#*
#*
#*             contact nepenthesdev@gmail.com
#*
#*********************************************************************************/
# mapping socket -> attackid
#  self.dbh = sqlite3.connect(user = g_dionaea.config()['modules']['python']['logsql']['file'])
```

```
#   self.cursor.execute("""CREATE TABLE IF NOT EXISTS
#   bistreams (
#    bistream INTEGER PRIMARY KEY,
#    connection INTEGER,
#    bistream_data TEXT
#   )""")
#
#  self.cursor.execute("""CREATE TABLE IF NOT EXISTS
#   smbs (
#    smb INTEGER PRIMARY KEY,
#    connection INTEGER,
#    smb_direction TEXT,
#    smb_action TEXT,
#    CONSTRAINT smb_connection_fkey FOREIGN KEY (connection) REFERENCES connections (connection)
#   )""")
#     print("dcerpcservice %s existed %s " % (servicecls.uuid, e) )
#  print(r)
#  print(r)
#  print(r)
#      print("%s %s %s %s %s existed" % (dcerpcservice, uuid, name, op, vuln))
# NetPathCompare was called NetCompare in dcerpcserviceops
# fix a typo on emu_services table definition
# emu_services.emu_serive is wrong, should be emu_services.emu_service
# 1) rename table, create the proper table
# 2) copy all values to proper table, drop old table
# fix a type on downloads table definition
# downloads.downloads is wrong, should be downloads.download
# 1) rename table, create the proper table
#   print(e)
# 2) copy all values to proper table, drop old table
#  self.cursor.execute("""CREATE TABLE IF NOT EXISTS
#   httpheaders (
#    httpheader INTEGER PRIMARY KEY,
```

```
#    connection INTEGER,
#    http_headerkey TEXT,
#    http_headervalue TEXT,
#    --  CONSTRAINT  httpheaders_connection_fkey  FOREIGN  KEY  (connection)  REFERENCES  connections
(connection)
#   )""")
#
#  for idx in ["headerkey","headervalue"]:
#   self.cursor.execute("""CREATE INDEX IF NOT EXISTS httpheaders_%s_idx
#   ON httpheaders (httpheader_%s)""" % (idx, idx))
# connection index for all
# updates, database schema corrections for old versions
# svn rev 2143 removed the table dcerpcs
# and created the table dcerpcrequests
#
# copy the data to the new table dcerpcrequests
# drop the old table
#   print(e)
#  print("unknown")
# maybe this was a early connection?
# the connection was linked before we knew it
# that means we have to
# - update the connection_root and connection_parent for all connections which had the pending
# - update the connection_root for all connections which had the 'childid' as connection_root
# Set the ID table ready for Logstash
# if we have to link a connection with a connection we do not know yet,
# we store the unknown connection in self.pending and associate the childs id with it
# if the new accepted connection was pending
# assign the connection_root to all connections which have been waiting for this connection
# not detected = '' -> NULL
#    logger.debug("scanner {} result {}".format(av,scans[av]))
```

File: settings.py

Norsk forklaring av kode:

Django settings for DionaeaFR project.

('Your Name', 'your_email@example.com'),

How many days (going backwards) worth of results to show

Local time zone for this installation. Choices can be found here:

http://en.wikipedia.org/wiki/List_of_tz_zones_by_name

although not all choices may be available on all operating systems.

On Unix systems, a value of None will cause Django to use the same

timezone as the operating system.

If running in a Windows environment this must be set to the same as your

system time zone.

Language code for this installation. All choices can be found here:

http://www.i18nguy.com/unicode/language-identifiers.html

If you set this to False, Django will make some optimizations so as not

to load the internationalization machinery.

If you set this to False, Django will not format dates, numbers and

calendars according to the current locale.

If you set this to False, Django will not use timezone-aware datetimes.

Absolute filesystem path to the directory that will hold user-uploaded files.

Example: "/home/media/media.lawrence.com/media/"

URL that handles the media served from MEDIA_ROOT. Make sure to use a

trailing slash.

Examples: "http://media.lawrence.com/media/", "http://example.com/media/"

Absolute path to the directory static files should be collected to.

Don't put anything in this directory yourself; store your static files

in apps' "static/" subdirectories and in STATICFILES_DIRS.

Example: "/home/media/media.lawrence.com/static/"

URL prefix for static files.

Example: "http://media.lawrence.com/static/"

Additional locations of static files

List of finder classes that know how to find static files in

various locations.

Make this unique, and don't share it with anybody.

```
# List of callables that know how to import templates from various sources.
# 'django.template.loaders.eggs.Loader',
# Python dotted path to the WSGI application used by Django's runserver.
# A sample logging configuration. The only tangible logging
# performed by this configuration is to send an email to
# the site admins on every HTTP 500 error when DEBUG=False.
# See http://docs.djangoproject.com/en/dev/topics/logging for
# more details on how to customize your logging configuration.
```

File: csv2sqlite.py

Norsk forklaring av kode:

```
#!/usr/bin/env python
#
# create a sqlite database from a csv file
# creates table schema and inserts rows
# can handle multiple csv files
#
# ./csv2sqlite a.csv bs.csv
# will create tables a and bs and bs will get the primary key of type integer "b"
#
```

File: gnuplotsql.py

Norsk forklaring av kode:

```
#!/usr/bin/python3
# create list of *all* days
# round start and stop by month
# create a list of ranges
# (overview|year|month,start,stop)
# create directories
#   print(path)
# create index.html files
# Years
# write months
```

Overviews

#print(db_query)

fill with zeros

write data file

general overview

protocols

File: logsql2postgres.py

Norsk forklaring av kode:

#!/opt/dionaea/bin/python3

sudo su postgres

createdb --owner=xmpp logsql

psql -U xmpp logsql < modules/python/util/xmpp/pg_schema.sql

print("{0} {1} {2}".format(offset, limit, r))

update the sequence if we inserted rows

FIXME postgres does not know connection_type pending

connection_type is an enum, so this may get messy

x

db['pg']['dbh'].commit()

File: readlogsqltree.py

Norsk forklaring av kode:

#!/opt/dionaea/bin/python3.1

args

print(connections)

File: retry.py

Norsk forklaring av kode:

#!/opt/dionaea/bin/python3.1

File: updateccs.py

Norsk forklaring av kode:

#!/opt/dionaea/bin/python3

#

#

Basing on:

gencc: A simple program to generate credit card numbers that pass the MOD 10 check

(Luhn formula).

Usefull for testing e-commerce sites during development.

#

#

http://www.darkcoding.net/credit-card-generator/

#

generate digits

Calculate sum

Calculate check digit

File: pg_backend.py

Norsk forklaring av kode:

#!/usr/bin/python -u

#

```
# aptitude install python-pyxmpp python-pgsql
#
# with db
#  ./pg_backend.py -U USER@sensors.carnivore.it -P XMPPPASS -M dionaea.sensors.carnivore.it -C anon-files -C anon-events -s DBHOST -u DBUSER -d xmpp -p DBPASS -f /tmp/
#
# without db
#  ./pg_backend.py -U USER@sensors.carnivore.it -P XMPPPASS -M dionaea.sensors.carnivore.it -C anon-files -C anon-events -f /tmp/
# PyXMPP uses `logging` module for its debug output
# applications should set it up as needed
# libxml2 is cruel
# check if we have dionaea entries in the message
# provide a namespace ...
# I love xml namespaces ...
# dionaea
# rename the namespace for the dionaea entries
# get the incident
# use the incidents name to get the appropriate handler
#   method = self.handle_incident_debug
#     print("c: '%s'" % c)
# call the handler with the object
#       print(mname)
#   else:
#    print("method %s is not implemented" % mname)
#    self.handle_incident_not_implemented(user, stanza)
# kippo
#   print(d)
#    self.handle_incident_not_implemented(user, stanza)
# dionaea
#   print(addr)
#   print(_from)
#   print(to)
```

```
#   print(via)
#   print(sdp)
# if bare JID is provided add a resource -- it is required
# setup client with provided connection information
# and identity data
# register features to be announced via Service Discovery
# set up handlers for supported <iq/> queries
# set up handlers for <presence/> stanzas
# set up handler for <message stanza>
# XMPP protocol is Unicode-based to properly display data received
# _must_ convert it to local encoding or UnicodeException may be raised
# Component class provides basic "main loop" for the applitation
# Though, most applications would need to have their own loop and call
# component.stream.loop_iter() from it whenever an event on
# component.stream.fileno() occurs.
# vi: sts=4 et sw=4
```

File: conpot.singlelogline.py

Norsk forklaring av kode:

```
#!/usr/bin/env python
#
# Get conpot events from mysql database and print to logfile
#  Uses a temp file to keep track of last printed id
#
# Configuration:
#       Change LAST_CONNECTION_FILE, SQLITE_DB and database connection settings
#       Leave SQLITE_DB empty for mysql-db
#       Change LOGFILE or leave empty for output to screen
#       Change honeypot-network definitions
#
# Koen Van Impe
#   koen.vanimpe@cudeso.be      @cudeso         http://www.vanimpe.eu
#   20141210
```

#

File: dionaea-singlelogline.py

Norsk forklaring av kode:

#!/usr/bin/env python

#

Get dionaea events from sqlite database and print to logfile

Uses a temp file to keep track of last printed id

#

Configuration:

Change SQLITE_DB and LAST_CONNECTION_FILE

Change LOGFILE or leave empty for output to screen

#

Koen Van Impe

koen.vanimpe@cudeso.be @cudeso http://www.vanimpe.eu

20141206

#

File: glastopf-singlelogline.py

Norsk forklaring av kode:

#!/usr/bin/env python

#

Get glastopf events from mysql database and print to logfile

Uses a temp file to keep track of last printed id

#

Configuration:

Change LAST_CONNECTION_FILE, SQLITE_DB and database connection settings

Leave SQLITE_DB empty for mysql-db

Change LOGFILE or leave empty for output to screen

Change honeypot-network definitions (DSTP, DSTPORT, PROTOCOL)

#

Koen Van Impe

koen.vanimpe@cudeso.be @cudeso http://www.vanimpe.eu

20141210

#

File: query_ELK.py

Norsk forklaring av kode:

#!/usr/bin/python

File: inspect-to-csv.py

Norsk forklaring av kode:

#!/usr/bin/env python

#

Process the 'Inspect' command from Kibana and convert to CSV

#

Save the 'inspect' window output in the variable 'request_file'

It will extract the URL and request and return CSV output

Does not work on histogram ...

#

Koen Van Impe on 2014-12-31

koen dot vanimpe at cudeso dot be

license New BSD : http://www.vanimpe.eu/license

#

#

Read the request

Split URL and request

Get the response from the Elasticsearch server

Print out all the master elements

#for el in response_json:

print el

Print the full response

#print response.text