Plest - quick and dirty graylisting of files

Getting started

Before you begin, start by prepapring nsrllookup. This commando will take a while.

```
docker-compose -f settings/svr/docker-compose-prepare.yml up svr-prepare &&
docker-compose -f settings/svr/docker-compose-prepare.yml rm -fsv
```

To download and start all services, do:

```
docker-compose up
```

This will bring everything up. Note that the Plaso container does not have a proper run command, so it will immediately stop. It's only in the docker-compose definition in order to be downloaded.

Also note that it takes a while for the nsrllookup service to start.

To set the passwords for the Elasticsearch cluster, run:

```
docker exec -it plaso-es ./bin/elasticsearch-setup-passwords interactive
```

Then change the password in `settings/kibana/kibana.yml` to the appropriate password. Restart the Kibana container:

```
docker-compose restart kibana
```

Running Plaso

Extract data, including hashes, from `testdata/evidences/`.

```
docker run -v ${pwd}/testdata/:/data log2timeline/plaso log2timeline --hashers all
/data/evidences.plaso /data/evidences
```

Enrich the data with data from nsrlsvr.

```
docker run --network plest_default -v ${pwd}/testdata/:/data log2timeline/plaso
psort --analysis nsrlsvr --nsrlsvr-hash md5 --nsrlsvr-host svr --nsrlsvr-port 9120
-o null /data/evidences.plaso
```

Write the data to timeline.log. Use `-o elasticsearch` to output data to Elasticsearch instead.

```
docker run -v ${pwd}/testdata/:/data log2timeline/plaso psort -w
/data/timeline.log /data/evidences.plaso
```

NSRLlookup

NSRLlookup is based on nsrllookup by cybagard.