eCBL to eXBL - Instructions

eCBL will be deprecated March 2021

PIPELINE
ACTIONABLE CYBER THREAT DATA

eCBL to eXBL - Instructions

Our new Extended XBL (eXBL) dataset, a metadata-enriched version of the XBL list, replaces the previous "Extended CBL" (ECBL) file ("cbl.diagnostics") that you have been getting so far using rsync.

eXBL comes as a JSON file called **"exbl_v4.json"** (a similar IPv6 file may follow later on) where each line is a record containing data associated with bot traffic relative to an IPv4 address. The fields are named, and they are described in https://docs.spamhaustech.com/extended-data/docs/source/03-datasets/010-exbl.html. We believe that the new format makes it easier to analyze bot data and is in line with current industry practices.

The size of the file is currently around 2.7 GB, and it is produced every hour. Of course, we strongly suggest to also rsync it every hour, to track variations as they occur and also to prevent an excessive accumulation of changes that would then make rsyncing times longer.

After some testing, we decided to make this file available both as a plain file, and as a gzip-compressed file (whose size is currently around 330 MB).

The compressed version is much smaller, but compression renders the rsync algorithm ineffective so that essentially the whole compressed file must be transmitted every time. Our tests indicate that installations with good network latencies and connectivity with at least one of our rsync servers - which we believe to be the majority - should be better served by rsyncing the plain file: updates normally imply a transfer of about 50 MB in both directions. In contrast, installations that experience high latencies with our servers due to their geographical location may be better served by rsyncing the compressed file instead, and then uncompressing it locally.

We have constructed a synchronization script to simplify the task of dealing with this choice, to make synchronization resilient against failures, and to assist with logs and diagnostics in case something goes wrong.

We kindly ask all customers to synchronize eXBL exclusively by using this script, available via rsync (from an authorized IP) at the URL:

```
rsync://na.dr.spamhaus.net/tools/spamhaus-exbl-sync.sh
```

After downloading the script, you need to edit it and properly define the following two variables in the "CUSTOMER CONFIGURATION" section at the beginning of the file:

```
SPAMHAUSDIR=XXXXX
```

You need to put the full path name of the directory you will associate with our service. This may well be the directory already in use for this purpose.

```
POOL="XX.dr.spamhaus.net"
```

"XX" must be replaced with
- "na" to select servers in North America
- "eu" to select servers in Europe
- "oc" to select servers in Oceania/South-East Asia

You can optionally put the IP address of a rsync server close to you in the PREFERRED variable. This will optimize the transfer times but it is not strictly necessary. If the preferred server is not available for any reason, the script will use one of the other servers in the selected pool. You can download and run the utility at rsync://na.dr.spamhaus.net/tools/rsync-servers-rtts.sh to find the best server for you.

You can also optionally enable compressed transfers - with the caveats discussed above - by setting GZIP=1 (default is GZIP=0).

And you can also optionally disable checksum verification of the file (which we do **not** recommend but it saves some time, and rsync itself does something similar) by setting VERIFY=0 (default is VERIFY=1).

After you have configured the script, you can instruct your cron facility to execute it every hour, selecting a minute of the hour between 50 and 59. If data freshness is extremely critical for you, feel free to choose 50. Otherwise, we would appreciate if you could select a higher minute, as it helps us to spread the load on systems and lines. Do not try to run this script as root, as the script would refuse to execute for security reasons.

That should be all you need: the file should appear in $SPAMHAUSDIR and should be updated every hour.

A final recommendation: please make sure that the process consuming the data does not move the file away from the $SPAMHAUSDIR directory: its presence is required by the rsync algorithm to find the differences and transmit only those. Therefore, if you need the file elsewhere please make a copy or use links.

eXBL Rsync Installation

Note: this is the general "installation from scratch" text to be made available to new user of eXBL/rsync.

It does not contain a description of the dataset contents; it only describes how to set up the file synchronization.

eXBL comes as two JSON files called "exbl_v4.json" and "exbl_v6.json" for IPv4 and IPv6 data respectively.

Each line in these files is a record containing data associated with bot traffic relative to an IP address. The fields are named, and they are described in https://docs.spamhaustech.com/extended-data/docs/source/03-datasets/010-exbl.html We believe that the new format makes it easier to analyze bot data and is in line with current industry practices.

Accessing these files through rsync requires eXBL authorization to be granted for the specific IP addresses of your servers.

These files are generated on a hourly basis, and we strongly suggest to also rsync them every hour, to track variations as they occur and also to prevent an excessive accumulation of changes that would then make rsyncing times longer.

We have constructed a synchronization script to easy the task of dealing with this choice, to make synchronization resilient against failures, and to assist with logs and diagnostics in case something goes wrong. We kindly ask all customers to synchronize eXBL exclusively by using this script, available via rsync (from an authorized IP) at the URL

```
rsync://na.dr.spamhaus.net/tools/spamhaus-exbl-sync.sh
```

Configuration

After downloading the script, you need to edit it and properly define the following two variables in the "CUSTOMER CONFIGURATION" section at the beginning of the file:

```
* SPAMHAUSDIR=XXXXX
```

You need to put the full path name of the directory you will associate with our service. This may well be the directory already in use for this purpose.

```
* POOL="XX.dr.spamhaus.net"
```

"XX" must be replaced with
- "na" to select servers in North America
- "eu" to select servers in Europe
- "oc" to select servers in Oceania/South-East Asia

```
* PREFERRED=X.X.X.X
```

You can optionally put the IP address of a rsync server close to you in the PREFERRED variable. This will optimize the transfer times but it is not strictly necessary. If the preferred server is not available for any reason, the script will use one of the other servers in the selected pool. You can download and run the utility at rsync://na.dr.spamhaus.net/tools/rsync-servers-rtts.sh to find the best server for you.

```
* GZIP=0 or GZIP=1
```

After some testing, we decided to make available these files both as plain files, and as gzip-compressed files. The compressed versions are much smaller, but compression renders the rsync algorithm ineffective so that essentially the whole compressed file must be transmitted every time. You can enable compressed transfers by setting GZIP=1 (the default is GZIP=0).

Our tests indicate that installations with good network latencies and connectivity with at least one of our rsync servers - which we believe to be the majority - should be better served by rsyncing plain files: updates normally imply a transfer of the order 50 MB in both directions. In contrast, installations that experience high latencies with our servers due to their geographical location may be better served by rsyncing compressed files instead, and then uncompressing them locally.

```
* VERIFY=0 or VERIFY=1
```

You can also optionally disable checksum verification of the file (which we do not recommend but it saves some time, and rsync itself does something similar) by setting VERIFY=0 (default is VERIFY=1).

* If you need only IPv4 data you can save some bandwidth by disabling rsyncing of IPv6 data by setting

```
GET_IPV6_DATA=0
```

(or GET_IPV4_DATA=0 if you only need IPv6 data).

Execution via cron

After you have configured the script, you can instruct your cron facility to execute it every hour, selecting a minute of the hour between 50 and 59. If data freshness is extremely critical for you, feel free to choose 50. Otherwise, we would appreciate if you could select a higher minute, as it helps us to spread the load on systems and lines. Do not try to run this script as root, as the script would refuse to execute for security reasons.

That should be all you need: the file should appear in the directory indicated in $SPAMHAUSDIR and should be updated every hour.

A final recommendation: please make sure that the process consuming the data does not move the file away from the $SPAMHAUSDIR directory: its presence is required by the rsync algorithm to find the differences and transmit only those. Therefore, if you need the file elsewhere please make a copy or use links.

Logging

The script leaves a diagnostic output of the session in $SPAMHAUSDIR/Work/spamhaus-exbl-sync.out that must be sent to our support in case of problems. Previous outputs going back to 24 hours are saved using the classic rotation mechanism appending ".0", ".1", ".2", etc suffixes to the file name.

Log files consisting of one line for each transfer are also saved in

```
$SPAMHAUSDIR/Logs///
      -exbl_v4.log and
      $SPAMHAUSDIR/Logs///
      -exbl_v6.log .
```

This organization allows for easy clean up of old data.

Each line in the log files contains 16 fields separated by spaces.

They are in the order:

1. The time at which the rsync synchronization was started in ISO 8601 format
2. The IP address of the Spamhaus rsync server that served the request
3. The generation time embedded within the file in Epoch format
4. The value of the $GZIP flag (0 or 1)
5. The value of the $VERIFY flag (0 or 1)
6. The rsync check-sum block size used (rsync(1) --block-size / -B option)
7. The size of the plain file
8. The size of the rsynced file (which may be the compressed one if $GZIP=1)
9. The bytes sent (from you to the rsync server) during the rsync transfer
10. The bytes received (from the rsync server to you) during the rsync transfer
11. The total elapsed time in seconds
12. The elapsed time taken by rsync in seconds
13. The elapsed time taken by gunzip in seconds (it will be 0 if $GZIP=0)
14. The elapsed time taken by the checksum verification in seconds (it will be 0 if $VERIFY=0)

15. The number of retries. It should be 0. A consistent 1 probably means that the IP indicated in $PREFERRED is not in production - contact us for a suitable replacement.
16. Failure code, it must be 0. A nonzero code is a rsync failure code if positive, while -1 indicates that the checksum verification failed.

Support

Feel free to contact us for any clarification with respect to the operation of the script, the rsync service or the data.

Pipeline Security Customer Support Portal:
https://support.pipelinesecurity.net

Or simply email us at techsupport@pipelinesecurity.net

If you are experiencing troubles, please include the log file of the day and the output saved in Work/spamhaus-exbl-sync.out.