NTP Statistics Add-on for Splunk

A Technology Add-on for Splunk

Author: Frank Wayne

Version: 2.0.1

Table of Contents

Overview

About the NTP Statistics Technology Add-ons

The NTP reference implementation[1] (NTP) is the official software that implements network time protocol[2] services. NTP allows for monitoring using a variety of "stats" logs. Although the data are available in simple text format, the events contain timestamps unsupported by Splunk's timestamp parser. This leaves events indexed without accurate timestamps and makes it necessary to extract and calculate the actual timestamps at search time.

The NTP Statistics Add-on for Splunk (the Add-on), either for *nix or for Windows, allows a Splunk® Enterprise administrator to index NTP v4 monitoring log data with proper timestamps, and perform detailed field extractions and calculations. If you have more than a casual interest in NTP, this Add-on will help you analyze NTP monitoring sets with greater ease than was ever before possible, whether you have one NTP host or a thousand.

The Add-on

The Add-on consists of a series of inputs, extractions and calculations for several NTP source types. There are separate Add-ons for *nix and Windows. This documentation describes both varieties of the Add-on, though it normally refers to simply *the Add-on*. The two Add-ons are:

> NTP Statistics Add-on for Splunk for *nix (`TA_ntp_nix`)
> NTP Statistics Add-on for Splunk for Windows (`TA_ntp_win`)

The Index

Events are sent to the index configured as default. If you have a different index for NTP events, override the default in `local/inputs.conf`.

Source types

Several source types correspond to the different NTP stats logs.

- `ntp:clockstats`
- `ntp:cryptostats`
- `ntp:loopstats`
- `ntp:peerstats`
- `ntp:rawstats`
- `ntp:sysstats`
- `ntp:timingstats`

For a detailed description of these data, please see the Data Types section.

[1] http://www.ntp.org.
[2] RFC 5905 describes NTP v4 and is available at http://www.rfc-editor.org/info/rfc5905.

Release Notes

Version	Date	Description
1.0.2	2016-05-08	First public release.
2.0.0	2020-10-22	Fixed configuration file typos and converted to index-time timestamp calculation.
2.0.1	2020-10-28	Added support for clockstats NTP header variables. Added support for GNGGA NMEA strings. Added automatic lookups for NTP mode, stratum, leap warnings and reference IDs. Fixed rawstats time interpretations and converted timestamps to UNIX (Splunk) time.

Support

This Add-on is not supported by Splunk. For help or to report problems, please contact the author, Frank Wayne, at frank.wayne@northwestern.edu.

Important Changes in Version 2

No Scripts

Version 2 does not use scripted inputs.

Version 1 of this Add-on used scripts (either Python or PowerShell) to rewrite the event timestamp in the NTP logs before forwarding the events to the indexer. This was done because strftime(), and hence Splunk, does not understand the Modified Julian Day number as a date type. The events passed by the version 1 Add-on to the indexer had timestamps in ISO 8601 format. With index-time parsing, I am able to do the date conversion using INGEST_EVAL on the indexer, making version 2 much simpler, more portable, and less demanding of resources.

Backward Compatibility with Version 1 Event Formats

In case you had version 1 installed and have events in the rewritten format, the search-time parsing is compatible with version 1 and standard NTP event formats. Just replace the old Add-on and search will work as before, even for your old events.

Installation

Hardware and Software Prerequisites

The Add-on requires no special hardware or software requirements. If the system can run the Universal Forwarder, this Add-on should work.

Both Add-ons require a NTP v4 daemon (or service) running on the host that is configured to write monitoring logs to a specific location. This is discussed in the next section.

NTP Configuration

The NTP daemon or service (not the Add-on) gets its configuration from the `ntp.conf` file. This file has nothing to do with Splunk and resides separate from Splunk's directory structure. For the **Add-on for *nix**, it is at `/etc/ntp.conf`. For the **Add-on for Windows**, it is normally located in the installation directory under `etc\ntp.conf`. For example, the Meinberg port[3] of NTP on 64-bit Windows has its configuration at `C:\Program Files (x86)\NTP\etc\ntp.conf`. The file on 32-bit Windows would be at `C:\Program Files\NTP\etc\ntp.conf.`

In order for the Add-on to find NTP monitoring logs, (1) the NTP daemon must produce the monitoring sets (the `statistics` command must be present in `ntp.conf`) and (2) the monitoring sets must reside where the Add-on expects (specified by the `statsdir` command in `ntp.conf`). NTP does not generate monitoring logs by default and, unfortunately, there is no official location for them. (If you do not specify their location, they go somewhere, but this location apparently varies by build.) A popular location for *nix seems to be `/var/log/ntpstats`. For 64-bit Windows (based on the Meinberg port), the conventional location is `C:\Program Files (x86)\NTP\stats`. Therefore, the `ntp.conf` file (remember, a file unrelated to Splunk) used by NTP should contain either

```
statsdir /var/log/ntpstats
```

or

```
statsdir "C:\Program Files (x86)\NTP\stats\"
```

depending on the OS. If you are using a 32-bit Windows OS, the log directory will be `C:\Program Files\NTP\stats` instead.

In addition to specifying the location of the monitoring logs, you must specify the type(s) of monitoring logs that NTP generates. For instance, if you want loopstats and peerstats, `ntp.conf` must contain

```
statistics loopstats peerstats
```

If you change the log file naming defaults using the `filegen` command, the Add-on will not be able to find the log file. Normally, NTP creates monitoring logs that start with the type name and are followed by the date: e.g., `peerstats.20160101`. The files roll daily. If you must change the filenames from the default, then you will also have to create custom `inputs.conf` monitor inputs. This is not difficult and is described in the installation instructions.

Architecture

This Add-on should be installed on the NTP client host, where it provides inputs. It must also be installed on the indexer(s), where it will set the correct time stamp for the event using index-time evaluation. The Windows and *nix Add-ons have the same props.conf and transforms.conf files; install only the one Add-on appropriate to the Splunk server's operating system. Finally, the Add-on must be installed on the search head(s) to provide field extraction and a bunch of other information about each event. Again, install either the Windows or *nix version, as appropriate.

Splunk is normally unable to parse the timestamp on an NTP event. The format is the Modified Julian Day in one field and the UTC offset in seconds on that day in the next field. For example, an event in the NTP loopstats log looks like this:

```
57499 3898.248 0.005174694 33.676 0.006109855 0.000489 10
```

[3] https://www.meinbergglobal.com/english/sw/ntp.htm.

When the event arrives at the indexer, instead of using normal timestamp parsing, the Add-on uses an INGEST_EVAL expression to override the timestamp with a UTC time calculated from the event fields. The event is written to the index as-is, with no modification. (Note that this varies from version 1 of this Add-on.)

Installation Steps

Install the Add-on on Splunk Enterprise

Single-server Instance

If you have search and indexing on a single server, install the Add-on either directly from Splunkbase or by downloading, extracting and copying the Add-on to your `$SPLUNK_HOME/etc/apps` directory. (The Add-on is either `TA-ntp-nix` or `TA-ntp-win`.) No setup is required. If you want to customize index settings, create a `local/inputs.conf` and make your changes for each monitor stanza. Restart Splunk.

Distributed Search

If you have a separate search head, install the Add-on as described for the single-server case. In clustered environments, download, extract and copy the App into your `$SPLUNK_HOME/etc/shcluster/apps` directory and use the deployer to distribute the Add-on to your search head cluster.

Distributed Indexer

If you have a separate indexer, install the Add-on as described for a single-server installation and restart your indexer. If you have an indexer cluster, download, extract and copy the Add-on into the `$SPLUNK_HOME/etc/master` directory on your cluster master and push the bundle. Expect this to perform a rolling restart of your indexer cluster.

Install the Add-on on a Universal Forwarder

To install the Add-on, copy it to the `$SPLUNK_HOME/etc/apps` directory on the Universal Forwarder as you would with any other Add-on. To deploy it to forwarders using a deployment server, copy the Add-on to the `$SPLUNK_HOME/etc/deploymentapps` directory on the deployment server and perform the normal forwarder management tasks.

Before the Add-on can read log files you must enable the appropriate input stanzas. All the Add-on's inputs are disabled by default. You may enable whichever ones you need and modify the defaults. All input configuration is done by in `local/inputs.conf`.

If you want to use an index other than the default index, change the index for each monitor input.

*Notes for the Add-on for *nix*

The following are the preconfigured scripted input stanzas on *nix.

```
[monitor:///var/log/ntpstats/clockstats*]
[monitor:///var/log/ntpstats/cryptostats*]
[monitor:///var/log/ntpstats/loopstats*]
[monitor:///var/log/ntpstats/peerstats*]
[monitor:///var/log/ntpstats/protostats*]
[monitor:///var/log/ntpstats/rawstats*]
[monitor:///var/log/ntpstats/sysstats*]
[monitor:///var/log/ntpstats/timingstats*]
```

Notes for the Add-on for Windows

The Add-on for Windows contains twice the number of `inputs.conf` stanzas as the *nix Add-on because it addresses 32- and 64-bit Windows. The complete list of input stanzas for Windows follows.

```
[monitor://C:\Program Files\NTP\stats\clockstats*]
[monitor://C:\Program Files\NTP\stats\cryptostats*]
[monitor://C:\Program Files\NTP\stats\loopstats*]
[monitor://C:\Program Files\NTP\stats\peerstats*]
[monitor://C:\Program Files\NTP\stats\protostats*]
[monitor://C:\Program Files\NTP\stats\rawstats*]
[monitor://C:\Program Files\NTP\stats\sysstats*]
[monitor://C:\Program Files\NTP\stats\timingstats*]
[monitor://C:\Program Files (x86)\NTP\stats\clockstats*]
[monitor://C:\Program Files (x86)\NTP\stats\cryptostats*]
[monitor://C:\Program Files (x86)\NTP\stats\loopstats*]
[monitor://C:\Program Files (x86)\NTP\stats\peerstats*]
[monitor://C:\Program Files (x86)\NTP\stats\protostats*]
[monitor://C:\Program Files (x86)\NTP\stats\rawstats*]
[monitor://C:\Program Files (x86)\NTP\stats\timingstats*]
```

Only enable the scripted inputs your environment needs. If you have no 32-bit Windows servers, you need only enable “x86” inputs. Unless you have built an NTP daemon with debugging enabled, it cannot generate timingstats, so there is no need to enable that input. Likewise, clockstats are only generated for reference clocks. To determine what monitoring logs are useful to you, refer to the NTP documentation for monitoring set types.[4] Enabling unneeded stanzas will not break anything.

If you need to change the stats file location or prefix (because your environment uses `filegen` to change the default file prefix), you can create your own stanzas in lieu of or in addition to the provided ones. Do not change the stanzas in `default/inputs.conf`; create your custom stanzas in `local/inputs.conf` and leave the unneeded stanzas disabled. Make sure you set the correct `sourcetype` for your custom stanza. Event timestamps (`_time`) are derived from the events' Modified Julian Day numbers at index time for the eight NTP source types only, so the source type must be set correctly.

User Guide

Key Concepts

The Add-on provides a set of knowledge objects for NTP data. By installing the Add-on, an administrator can have access to various NTP monitoring statistics in a single place with all the analysis and visualization capabilities of Splunk.

A detailed description of the many statistics reported by NTP monitoring sets is beyond the scope of this document. Please refer to the official documentation for NTP[5] or David Mills' definitive book[6] for an explanation of these metrics.

[4] https://www.eecis.udel.edu/~mills/ntp/html/monopt.html#types.
[5] http://www.ntp.org/documentation.html
[6] https://www.eecis.udel.edu/~mills/book.html

Data Types

Each NTP monitoring set is associated with its own source type in Splunk. All source types are prefixed with `ntp:`. The following sections contain short descriptions of the fields extracted or calculated for each source type. Highlighted rows indicate calculated fields.

Where appropriate, fields that are in SI units have suffixes that correspond to SI units. These include `m` (meters), `s` (seconds), and `ms` (milliseconds). Splunk cannot use the Greek letter µ (*mu*) in a variable name, so `micros` is the alternate suffix used. When appropriate, fields expressing seconds are echoed in fields expressing milliseconds. For example, `clock_offset_s` and `clock_offset_ms` express the same offset at different scales. Milliseconds often make more sense in charts and visualizations and the author was tired of doing EVALs in every query.

ntp:clockstats

These are reference clock statistics and are available in NTP only when associated with a reference clock input, such as a GPS data stream or a PPS source.

```
57499 3898.248 127.127.4.1 93 226 00:08:29.606 D
```

ntp:clockstats		
Field	**Description**	**Example**
src_ip	Clock address	127.127.4.1
message	Some message	93 226 00:08:29.606 D

Some drivers will list NTP header variables in clockstats, like the information shown in rawstats.

```
59150 61223.943 192.168.0.5 192.168.0.221 3812893158.946794980 3812893158.947056383
3812893223.942137463 3812893223.942460073 0 4 3 2 6 -20 0.000351 0.003403
192.168.251.60
```

ntp:clockstats Header Data Fields		
Field	**Description**	**Example**
clock_offset_ms[7]	Difference between dest_ip and src_ip times (ms)	-0.030756
clock_offset_s[7]	Difference between dest_ip and src_ip times	-0.000030756
clock_source	IANA master clock source text	Generic pulse-per-second
dest_ip	Client address	192.168.0.221
leap_warning	Leap second warning indicator	0
leap_warning_text	Description of leap warning	no warning
mode	NTP packet mode	3
mode_text	NTP packet mode description	client
pbds_dst	Packet buffer data structure destination time (NTP epoch)	3812893223.942460073

[7] This is the *theta* value described on page 29 of RFC 5905. It is calculated only when all time fields are filled, i.e. when the reported packet is the received response.

ph_org	Packet header origin time (NTP epoch)	3812893158.946794980
ph_rec	Packet header receive time (NTP epoch)	3812893158.947056383
ph_xmt	Packet header transmit time (NTP epoch)	3812893223.942137463
poll_exponent	Polling internal exponent	6
poll_s	Polling internal	64
precision_exponent	Clock precision exponent	-20
precision_micros	Clock precision (µs)	0.95367431640625
ref_id	Reference ID	PPS
root_delay_ms	Clock root delay (ms)	0.351
root_delay_s	Clock root delay	0.000351
root_dispersion_ms[8]	Clock root dispersion (ms)	3.403
root_dispersion_s[8]	Clock root dispersion	0.003403
roundtrip_delay_ms[9]	Client to clock roundtrip time (ms)	0.584126
roundtrip_delay_s[9]	Client to clock roundtrip time	0.000584126
src_ip	Clock address	192.168.0.5
stratum	Clock stratum	1
stratum_text	Clock stratum description	primary server
time_destination[10]	UNIX epoch time of pbds_dst	1603904423.942460000
time_origin[10]	UNIX epoch time of ph_org	1603904358.946795000
time_receive[10]	UNIX epoch time of ph_rec	1603904358.947056300
time_transmit[10]	UNIX epoch time of ph_xmt	1603904423.942137200
version	NTP protocol version	4

If you happen to have a GPS sending NMEA GGA strings to NTP, these events will have more fields extracted. You must include GGA strings in your server mode setting (i.e., have bit 1 set) to have NTP record them in clockstats.

```
57499 3898.248 127.127.20.0
$GPGGA,211805.000,3908.4053,N,07713.1128,W,2,09,0.93,109.6,M,-34.1,M,0000,0000*5C
```

[8] Root dispersion is wildly inaccurate when the ph_dst is zero (and therefore the time_destination is null) and should be ignored in that circumstance.

[9] This is the *delta* value as described on page 29 of RFC 5905. It is calculated only when all time fields are filled, i.e. when the reported packet is the received response.

[10] All time_* fields are the UNIX epoch (Splunk) versions of the NTP epoch timestamps in the packet data.

ntp:clockstats NMEA GGA Additional Fields		
Field	**Description**	**Example**
gps_altitude_m	Altitude	109.6
gps_checksum	NMEA checksum	5C
gps_dgps_station_id	?	0000
gps_fix_quality	GPS fix quality	2
gps_fix_quality_text	GPS fix quality text	DGPS fix
gps_horizontal_dillution	Horizontal dilution of precision	0.93
gps_last_dgps_update_s	?	0000
gps_latitude	Latitude	39.140089
gps_latitude_direction	Latitude direction	N
gps_longitude	Longitude	-77.218546
gps_longitude_direction	Longitude direction	W
gps_tracked_satellites	Number of satellites tracked	09
gps_utc	GPS UTC clock	211805.000
gps_wgs84_m	Height of geoid above WGS84[11] ellipsoid	-34.1

The gps_latitude and gps_longitude are calculated fields, not direct extracts. The format of the NMEA location data is unusual; the data calculated by the App are in degrees, get a negative value for south and west, and are compatible with geolocation in Splunk. This Add-on makes it work right.

ntp:cryptostats

Autokey protocol events are recorded as the ntp:cryptostats source type. This monitoring is only available if the host uses cryptography.

```
57499 3898.248 192.168.0.128 82080150 23145 10e bad_or_missing_group key
```

ntp:cryptostats		
Field	**Description**	**Example**
src_ip	Source address	192.168.0.128
message	Log message	82080150 23145 10e bad_or_missing_group key

The src_ip is 0.0.0.0 when the event refers to the local system.

ntp:loopstats

The ntp:loopstats source type contains statistics related to the clock discipline loop. An event occurs when a clock update occurs. These data describe how close the clock is to UTC, how much its offset varies, how the system's oscillator varies from perfect clocking, as well as how that oscillator drifts due to temperature and other variables.

```
57499 3898.248 0.005629489 33.694 0.006250667 0.000682 10
```

[11] See https://en.wikipedia.org/wiki/World_Geodetic_System.

ntp:loopstats		
Field	**Description**	**Example**
`clock_offset_ms`	Clock offset in ms	`5.629489`
`clock_offset_s`	Clock offset	`0.005629489`
`frequency_offset_ppm`	Frequency offset	`33.694`
`jitter_ms`	Jitter in ms	`6.250667`
`jitter_s`	RMS clock jitter	`0.006250667`
`loop_time`	Clock discipline loop time	`1024`
`loop_time_exponent`	Clock discipline loop time exponent	`10`
`wander_ppm`	RMS frequency jitter	`0.000682`

ntp:peerstats

The `ntp:peerstats` source type reports peer statistics. An event occurs upon an update from either another NTP server or a reference clock. These data describe the other clock, including its offset and jitter, how far away it is (roundtrip delay), and whether it is being selected to determine local time.

```
57499 3898.248 112.118.251.24 9024 0.004007564 0.000976563 0.002408862 0.000976563
```

ntp:peerstats		
Field	**Description**	**Example**
`clock_offset_ms`	Clock offset in ms	`4.007564`
`clock_offset_s`	Clock offset	`0.004007564`
`dispersion_ms`	Dispersion in ms	`2.408862`
`dispersion_s`	Dispersion	`0.002408862`

`flags`	Flags present in the status word	`config`
		`reach`
		`sel_reject`
		`reachable`

`jitter_ms`	Jitter in ms	`0.976563`
`jitter_s`	Jitter	`0.000976563`
`peer_status_word`	Same as `status_word`	`9024`
`roundtrip_delay_ms`	Roundtrip delay in ms	`0.976563`
`roundtrip_delay_s`	Roundtrip delay	`0.000976563`
`src_ip`	Source IP	`112.118.251.24`
`status_count`	Count from the status word	`2`
`status_word`	Peer status word	`9024`

ntp:rawstats

The `ntp:rawstats` source type records timestamp statistics for response packets received by the host. (The destination is always an IP local to the host and the source address is the remote time source.) It includes the four timestamps in the NTP packet, which are expressed in UNIX epoch time. The `ref_id` can be the IP address of an upstream time source or a reference clock ID.

```
59150 84988.540 192.168.0.24 192.168.0.27 3812916988.539335999 3812916988.539468823
3812916988.539786274 3812916988.540085324 0 4 4 1 4 -19 0.000000 0.001022 .PPS.
```

ntp:rawstats

Field	Description	Example
clock_offset_ms[12]	Difference between dest_ip and src_ip times (ms)	-0.0832081
clock_offset_s[12]	Difference between dest_ip and src_ip times	-0.0000832081
clock_source	IANA master clock source text	Generic pulse-per-second
dest_ip	Destination address	192.168.0.27
leap_warning	Leap second warning	0
leap_warning_text	Description of leap warning	no warning
mode	Mode	4
mode_text	Mode text	server
pbds_dst	Packet buffer data structure destination time (NTP epoch)	3812916988.540085324
ph_org	Packet header origin time (NTP epoch)	3812916988.539335999
ph_rec	Packet header receive time (NTP epoch)	3812916988.539468823
ph_xmt	Packet header transmit time (NTP epoch)	3812916988.539786274
poll_exponent	Polling interval exponent	4
poll_s	Polling interval	16
precision_exponent	Source precision exponent	-19
precision_micros	Source precision (µs)	1.9073486328125
ref_id	Source clock reference ID	PPS
root_delay_ms	Clock root delay (ms)	0.000
root_delay_s	Clock root delay	0.000000
root_dispersion_ms	Clock dispersion (ms)	1.022
root_dispersion_s	Clock dispersion	0.001022
roundtrip_delay_ms[13]	Client to clock roundtrip time (ms)	0.431538
roundtrip_delay_s[13]	Client to clock roundtrip time	0.000431538
src_ip	Source IP	192.168.251.24
stratum	Source stratum	1
time_destination[14]	UNIX epoch time of pbds_dst	1603928188.540085300
time_origin[14]	UNIX epoch time of ph_org	1603928188.539336200
time_receive[14]	UNIX epoch time of ph_rec	1603928188.539468800
time_transmit[14]	UNIX epoch time of ph_xmt	1603928188.539786300
version	Source NTP version	4

[12] This is the *theta* value described on page 29 of RFC 5905. It is calculated only when all time fields are filled, i.e. when the reported packet is the received response.

[13] This is the *delta* value as described on page 29 of RFC 5905. It is calculated only when all time fields are filled, i.e. when the reported packet is the received response.

[14] All time_* fields are the UNIX epoch (Splunk) versions of the NTP epoch timestamps in the packet data.

ntp:sysstats

The ntp:sysstats is a record of system activity and perhaps the least verbose of all NTP statistics. One event occurs every hour.

```
57499 3898.248 3600 695 110 639 56 6 5 4 3 2 1
```

ntp:sysstats

Field	Description	Example
access_denied	Packets denied access	6
bad_auth	Packets failing authentication	4
bad_length_or_format	Packets incorrectly formatted	5
current_version	Packets of matching version	639
declined	Packets declined	3
kod_packets_out	Kiss-of-death packets sent	1
last_reset_s	Seconds since last counter reset	3600
old_version	Packets from obsolete version	56
Packets	Packets to this host	695
packets_generated	Packets from this host	110
rate_exceeded	Packets from hosts exceeding limits	2

ntp:timingstats

The ntp:timingstats source type collects process time data that is only available if an NTP daemon is compiled with process-time debugging enabled. References to the code path description are in the NTP source code.

```
57499 3898.248 10.0.0.1 1 0.000017229 input processing delay
```

ntp:timingstats

Field	Description	Example
src_ip	Server address	10.0.0.1
event_count	Event count	1
time_micros	Total processing time in µs	17.229
time_ms	Total processing time in ms	0.017229
time_s	Total processing time	0.000017229
Message	Code path description	input processing delay

Troubleshooting

Make sure NTP is configured to produce monitoring logs and the logs are present. Here is an example of entries in etc\ntp.conf (installed in C:\Program Files (x86)\NTP) on a 64-bit Windows server:

```
statsdir "C:\Program Files (x86)\NTP\stats\"
statistics loopstats peerstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
```

Make sure you have enabled the input for the log NTP is generating. Here is an example of a stanza in your `local/inputs.conf` that enables the *peerstats* monitor and sends the events to an index named *ntp*:

```
[monitor://C:\Program Files (x86)\NTP\stats\peerstats*]
disabled = false
index = ntp
```

Make sure the logs are going to the location the input expects. Make sure you create the log directory on Windows if it does not exist, otherwise logging may silently fail.

Make sure your indexer has one of the Add-ons installed.

Please note that this Add-on sets the Splunk timestamp of each event at index time. If you have old NTP log events ingested without this Add-on installed, those events will continue to have whatever timestamps were set at the time they were indexed.

Upgrade Instructions

Update the Add-on on the indexer(s). If the version 1 Add-on was not installed on your indexer(s), make sure to install this version so that the timestamps are set correctly.

Update the Add-on on your search head(s).

Configure the version 2 monitor inputs to provide the data that the version 1 scripted inputs provided. Deploy the Add-on to your NTP hosts.

Version 1 did not use file system monitors for inputs, but scripted inputs that kept their own version of the Splunk fish bucket (records of input files and positions). When you install version 2, it will reindex everything in your NTP log directories (as enabled in your `local/inputs.conf`). This means you will have duplicate records on your indexer(s). To minimize that, disable the version 1 Add-on, move or delete the log files on your NTP hosts, and then upgrade and enable the new version of this Add-on.