Using DNSTAP with BIND

Author: **Brian Reid** Reference Number: **AA-01342** Views: **23302** Created: **2016-01-18 20:15** **100 Rating**/ 1 Voters
Last Updated: **2018-04-20 23:07**

Introduction

dnstap is a fast, flexible method for capturing and logging DNS traffic. Developed by Robert Edmonds at Farsight Security, Inc., it is supported by several DNS implementations, including BIND. Some information about it can be found on its website at dnstap.info

dnstap will be generally available in BIND 9.11 but is in certain editions of earlier versions, such as BIND 9.9.8-S5.

- The open-source **dnstap** system is both a file format and software to create files in that format. To use it with BIND, three things must happen:
- BIND must be compiled with **dnstap** support included and configured to enable that support at runtime
- The BIND configuration file "options" statement can include four dnstap-specific options:
    - A required **dnstap** option that turns it on at runtime and specifies which message types are to be logged,
    - A required **dnstap-output** option that specifies what to do with the captured packets. It is more efficient to send the captured packets to a process waiting to receive them than to write them to a file, but there is a configuration option that will write to a file.
    - An optional dnstap-identity option that sets the string that will be used to label the captured data. If not specified the hostname will be used.
    - An optional dnstap-version option that specifies the version number associated with the captured data. If not specified, the BIND release version number will be used.
- Assuming that you have specified in your BIND configuration that the dnstap packets are to be sent to a waiting process, that process must exist and be listening to receive the captured packets that are sent. This process is part of the **dnstap** software, not part of BIND, and must be installed and run separately. It must be installed on the same server that is running BIND.

Installing and enabling dnstap and BIND with dnstap support

First ensure that the version of BIND you are running has dnstap support . BIND 9.9.8-S5 has such support. Check the release notes for other versions. **dnstap** support is not included in major releases until BIND 9.11.

Configuring BIND to use dnstap

Include "`--enable-dnstap`" in the build options. The exact method by which you can adjust the build options is OS-specific.

To build BIND with **dnstap** support you must ensure that the runtime libraries required by the **dnstap** sender (which will be embedded in BIND) are present. There are two nonstandard library packages that **dnstap** requires, and one of those depends in turn on a Google library package. These three packages are all available from github, and you must install them in this order:

https://github.com/google/protobuf
https://github.com/protobuf-c/protobuf-c
https://github.com/farsightsec/fstrm

To install, clone each repository onto your build machine:

```
git clone https://github.com/google/protobuf
git clone https://github.com/protobuf-c/protobuf-c
git clone https://github.com/farsightsec/fstrm
```

Then in each of those cloned directories, run

```
autoreconf -i
configure
make; make install
```

Once those libraries are installed on your build machine, build and install the BIND binary using the standard build process that is documented in the BIND 9 ARM. Note that the farsightsec/fstrm package will include the fstrm_capture program mentioned below.

Once the **dnstap**-enabled BIND is built, you need to adjust your configuration so that the running BIND process will actually use **dnstap**. In the options statement in the BIND configuration file, include a **dnstap** option that specifies the message types that you would like to have logged. The choices are `client`, `auth`, `resolver`, and `forwarder`. Each type can take an additional argument that indicates whether to log query messages or response messages; if not specified, both queries and responses are logged. The rest of the dnstap-related options can go next; only the **dnstap-output** option is required.

You could, for example, write

```
dnstap {auth; resolver query;};
dnstap-output unix "/var/run/bind/dnstap.sock"
```

Or, to capture the output directly in a file,

```
dnstap {auth; resolver query;};
dnstap-output file "/var/tmp/example.dnstap"
```

Or you could put in all of the possible options:

```
dnstap {auth; resolver query;};
dnstap-output unix "/var/run/bind/dnstap.sock"
dnstap-identity hostname
dnstap-version 9.9.8-S5
```

With those options specified, BIND will upon start or restart make a connection to the unix-domain socket named /var/run/bind/dnstap.sock, which must already exist if that connection is to succeed. The socket is created by running the dnstap listener program, which is named **fstrm_capture**. Properly installed, it will create a unix-domain socket /var/run/bind/dnstap.sock and listen to it. Any data that comes through will be saved to a file named example.dnstap

Starting the **fstrm_capture** program before starting BIND will ensure that the socket exists by the time BIND needs to find it and connect to it.

Installing fstrm_capture

The fstrm_capture program is part of the dnstap software distribution and not the BIND software distribution. Instructions for retrieving its source and installing it on your server can be found on its website http://dnstap.info . That website also documents other options and tools available as part of the dnstap software.

installing and building fstrm_capture may not be necessary as a separate step

Note that the farsightsec/fstrm package will include the fstrm_capture program and should build it automatically in most cases.

```
# fstrm_capture -t protobuf:dnstap.Dnstap -u /var/run/bind/dnstap.sock -w /var/tmp/example.dnstap
```

A recorded webinar on the use of dnstap is available from Men and Mice. While its examples are taken from other software than BIND, the principles are identical and the configuration and use of the listening software is identical.

BIND utility dnstap-read

Every BIND distribution that contains dnstap support includes the dnstap-read utility, which decodes dnstap files and outputs the result in a human-readable format. Its brief man page, which is installed as part of the BIND distribution, is reproduced below.

Name

dnstap-read — print dnstap data in human-readable form

Synopsis

dnstap-read [-m] [-p] [-y] {file}

DESCRIPTION

dnstap-read reads dnstap data from a specified file and prints it in ahuman-readable format. By default, dnstap data is printed in a shortsummary format, but if the -y option is specified, then a longer andmore detailed YAML format is used instead.

OPTIONS

-m Trace memory allocations; used for debugging memory leaks.

-p After printing the dnstap data, print the text form of theDNS message that was encapsulated in the dnstap frame.

-y Print dnstap data in a detailed YAML format. Implies -p.

SEE ALSO

named(8), nsupdate(8), BIND 9 Administrator Reference Manual.