Understanding The Nagios Network Analyzer 2026 Backend

Purpose

This document describes how the backend scripts of Nagios Network Analyzer 2026 work and provides the locations for data storage.

Data Location

Flow data collected by Nagios Network Analyzer is kept in the `/usr/local/nagiosna/var` directory for the default location. Within this `var/` directory, subfolders are created for each individual Source. The Source names get sanitized to make them filesystem safe and will show under `var/` in a form that is similar to their given name.

For instance, a Source called `This Has Spaces` gets changed to `ThisHasSpaces` and a new directory in `var/` is created. To access the flow information for `This Has Spaces`, you would access `var/ThisHasSpaces`. Inside this directory, there are three items of interest:

- `flows/` directory
    - The `flows/` directory contains the binary files that `nfcapd` produces. These are used to get granular NetFlow information using `nfdump`.
- `bandwidth.rrd`
    - The `bandwidth.rrd` is used to generate on-demand traffic graphs.

Data from the integrations of Nmap, Suricata and Wireshark and stored in separate locations. For data that is stored not in the database, they will be stored at these locations:

- Nmap
    - `/usr/local/nmap`
- Suricata configs
    - `/usr/local/etc/suricata`
- Suricata logs
    - `/usr/local/var/log/suricata`
- Suricata rules
    - `/usr/local/var/lib/suricata`
- Wireshark
    - `/usr/local/wireshark`

How Data Is Captured

Nagios Network Analyzer relies on `nfcapd`/`sfcapd` to capture flow data. Each individual Source has a process that is spawned for it to capture data. Each process opens up a port and listens on this port. Any valid NetFlow data it receives will be processed and recorded in the `flows/nfcapd.current` file and will be expired into a new file, `flows/nfcapd.<datestamp>`, whenever the system clock hits a 5-minute interval.

Once this new file has been made, a callback script gets kicked off by the `nfcapd` process. This callback script, `reap_files.py`, is in charge of updating the `bandwidth.rrd` for the Source.

This callback script runs every 5 minutes, or every time a new `nfcapd` file is made. We cannot stress the importance of this concept enough when troubleshooting or adding functionality to Nagios Network Analyzer. This callback script logs to `nagiosna/var/backend.log`.

The bottom line: If information is updated, or some action thhappens based on the arrival of new NetFlow data, it happens somewhere in the callback script.

Also, on the creation of a new `nfcap` file, `nfexpire` is called initially to clean up any old files. `nfexpire` is independent of the callback script and will remove `nfcap` files that are older than the data lifetime that is specified at source creation.

Note: Previous versions of NNA also ran checks through `reap_files.py`, but in Network Analyzer 2026 they are run through Laravel jobs.

We won't go into discussion for data collection with Nmap, Suricata and Wireshark as each of these integrations use their base functionality to perform their tasks. If you want to learn more about the data collection functions of these integrations please view their official documentation.

How Bandwidth RRDs Are Updated

The bandwidth RRDs hold how much traffic has been observed at each update interval. They hold information about Bytes, Flows, Packets, Bytes/Sec, Packets/Sec and Bytes/Packet. These RRDs are updated at each 5-minute interval and are numbers that are derived from running `nfdump` on all newly created `nfcap` files. The callback script `reap_files.py` is how the RRD files are updated.

General Troubleshooting Advice

The heartbeat of Nagios Network Analyzer is the `nfcapd` process. Keep in mind, each Source gets its own `nfcapd`/`sfcapd` process. So, when troubleshooting issues arise with the backend, it is generally best to make sure that the Source-in-question's `nfcapd` is running. This is done by logging onto the Network Analyzer server as the root user and grepping through the ps output from the command line:

```
ps aux | grep <directory of source>
```

If this isn't running, then the problem begins there, and can be remedied by restarting the process via:

```
/usr/local/nagiosna/bin/nagiosna restart
```

Note: You will reference the name of the directory for the Source in question and not the name of the Source in Network Analyzer.

If you want to start a specific source run this command:

```
sudo -u nna python3 /usr/local/nagiosna/scripts/source_controller.py
start <Source Name>
```

If you want to stop a specific source run this command:

```
sudo -u nna python3 /usr/local/nagiosna/scripts/source_controller.py stop
<Source Name>
```

Once you have verified that `nfcapd` is indeed running for the Source in question, it is time to check to make sure that there is nothing weird showing up in the `var/backend.log`. Any known failures that are occurring that can cause a host of issues will show up there. If the callback script ever has issues, it will be logged here with the reason.

Another major technique that will be very helpful is determining whether or not traffic is being received by the Nagios Network Analyzer server on a specific port. This is very easily done via the use of `tcpdump`. This will absolutely tell you whether or not NetFlow traffic is coming in. Simply run the command:

```
tcpdump dst port <port that traffic is supposed to be coming in on>
```

If there is no traffic reported here, you'll need to go back to the device that is allegedly sending NetFlow data. If you see the traffic reported, then you may also want to check that the local firewall has the correct rules created. These are created automatically by Network Analyzer but can be checked by running:

```
iptables –L
```

Or if your system uses firewalld you can run:

```
firewall-cmd –list-all
```

Some `nfdump` commands are helpful for troubleshooting incoming NetFlow traffic. To ensure that data is actually coming from the NetFlow device and is being captured by the `nfcap` process, navigate to the `flows/` directory of a Source in question and find the newest `nfcap` file that is not `nfcap.current` and run:

```
nfdump -r <newest nfcap file>
```

If this shows any data at all, it should also be present in the web interface, if this shows zeros, then the web interface should be showing zeros as well.

Finishing Up

This completes the documentation on Understanding the Nagios Network Analyzer 2026 backend. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum Visit Nagios Knowledge Base Visit Nagios Library