DNP3 Inspector

- DNP3 Inspector Overview, on page 1
- DNP3 Inspector Parameters, on page 1
- DNP3 Inspector Rules, on page 2
- DNP3 Inspector Intrusion Rule Options, on page 2

DNP3 Inspector Overview

Type	Inspector (service)
Usage	Inspect
Instance Type	Multiton
Other Inspectors Required	`stream_tcp`, `stream_udp`
Enabled	`false`

Distributed Network Protocol (DNP3) is a Supervisory Control and Data Acquisition (SCADA) protocol that was originally developed to provide consistent communication between electrical stations. DNP3 is widely used in the water, waste, and transportation industries.

The `dnp3` inspector detects anomalies in DNP3 traffic and analyzes the DNP3 protocol. The `dnp3` intrusion rule options access certain DNP3 protocol fields.

DNP3 Inspector Parameters

DNP3 TCP port configuration

The `binder` inspector defines the DNP3 TCP port configuration. For more information, see the Binder Inspector Overview.

Example:

```
[
   {
       "when": {
           "role": "any",
           "service": "dnp3"
    },
       "use": {
           "type": "dnp3"
       }
    }
]
```

check_crc

Specifies whether to validate the checksums contained in DNP3 Link-Layer Frames. The `dnp3` inspector ignores frames with invalid checksums. If intrusion rule 145:1 is enabled, Snort generates alerts for invalid checksums.

Type: boolean

Valid values: `true`, `false`

Default value: `false`

DNP3 Inspector Rules

Enable the `dnp3` inspector rules to generate events and, in an inline deployment, drop offending packets.

Table 1: DNP3 Inspector Rules

GID:SID	Rule Message
145:1	DNP3 link-layer frame contains bad CRC
145:2	DNP3 link-layer frame was dropped
145:3	DNP3 transport-layer segment was dropped during reassembly
145:4	DNP3 reassembly buffer was cleared without reassembling a complete message
145:5	DNP3 link-layer frame uses a reserved address
145:6	DNP3 application-layer fragment uses a reserved function code

DNP3 Inspector Intrusion Rule Options

dnp3_data

The `dnp3_data` keyword positions the detection cursor to the beginning of the DNP3 data in an application layer fragment, regardless of preceding rule options. With this option you can write rules based on the data within fragments without splitting up the data and adding CRCs every 16 bytes.

Syntax: `dnp3_data;`

Examples: `dnp3_data;`

dnp3_func

This option matches against the function code inside a DNP3 application layer request/response header. The code may be a decimal number or a string from the list below.

Type: string

Syntax: `dnp3_func: <DNP3_function>;`

Valid values: *`DNP3_function`* is one of the following:

- An integer from `0` to `255`
- `confirm` (Corresponds to function code `0`.)
- `read` (Corresponds to function code `1`.)
- `write` (Corresponds to function code `2`.)
- `select` (Corresponds to function code `3`.)
- `operate` (Corresponds to function code `4`.)
- `direct_operate` (Corresponds to function code `5`.)
- `direct_operat_nr` (Corresponds to function code `6`.)
- `immed_freeze` (Corresponds to function code `7`.)
- `immed_freeze_nr` (Corresponds to function code `8`.)
- `freeze_clear` (Corresponds to function code `9`.)
- `freeze_clear_nr` (Corresponds to function code `10`.)
- `freeze_at_time` (Corresponds to function code `11`.)
- `freeze_at_time_nr` (Corresponds to function code `12`.)
- `cold_restart` (Corresponds to function code `13`.)
- `warm_restart` (Corresponds to function code `14`.)
- `initialize_data` (Corresponds to function code `15`.)
- `initialize_appl` (Corresponds to function code `16`.)
- `start_appl` (Corresponds to function code `17`.)
- `stop_appl` (Corresponds to function code `18`.)
- `save_config` (Corresponds to function code `19`.)
- `enable_unsolicited` (Corresponds to function code `20`.)
- `disable_unsolicited` (Corresponds to function code `21`.)
- `assign_class` (Corresponds to function code `22`.)
- `delay_measure` (Corresponds to function code `23`.)
- `record_current_time` (Corresponds to function code `24`.)
- `open_file` (Corresponds to function code `25`.)

- `close_file` (Corresponds to function code `26`.)
- `delete_file` (Corresponds to function code `27`.)
- `get_file_info` (Corresponds to function code `28`.)
- `authenticate_file` (Corresponds to function code `29`.)
- `abort_file` (Corresponds to function code `30`.)
- `activate_config` (Corresponds to function code `31`.)
- `authenticate_req` (Corresponds to function code `32`.)
- `authenticate_err` (Corresponds to function code `33`.)
- `response` (Corresponds to function code `129`.)
- `unsolicited_response` (Corresponds to function code `130`.)
- `authenticate_resp` (Corresponds to function code `131`.)

Examples:

```
dnp3_func: 1;
```

```
dnp3_func: delete_file;
```

dnp3_ind

Provide a list of Internal Indicator flags to match against the Internal Indicator flags in a DNP3 application layer response header. If you provide multiple flags in one option, the rule fires if any one of the flags is set. To alert on multiple flags, use multiple rule options.

Type: string

Syntax: `dnp3_ind: "<flag> <flag>";`

Valid values: One or more DNP3 Internal Indicator flags where `flag` is one of the following:

- `all_stations`
- `class_1_events`
- `class_2_events`
- `class_3_events`
- `need_time`
- `local_control`
- `device_trouble`
- `device_restart`
- `no_func_code_support`
- `object_unknown`
- `parameter_error`

- `event_buffer_overflow`
- `already_executing`
- `config_corrupt`
- `reserved_2`
- `reserved_1`

Examples:

Alert on device restart OR on initiation of time synchronization:

```
dnp3_ind:"device_restart need_time";
```

Alert on class_1 AND class_2 AND class_3 events:

```
dnp3_ind:class_1_events; dnp3_ind:class_2_events; dnp3_ind:class_3_events;
```

dnp3_obj

Matches on DNP3 object header groups and variations.

Type: integer

Syntax: `dnp3_obj:<groupnum>,<varnum>;`

Valid values: DNP3 object group identifiers and variation identifiers, where:

- *`groupnum`* is an integer from `0` to `255` specifying a DNP3 object group.
- *`varnum`* is an integer from `0` to `255` specifying a variation within the object group.

Examples:

Alert on DNP3 `Date and Time` object:

```
dnp3_obj:50,1;
```