Meer User Guide Documentation

Release 0.0.3

Champ Clark III

Nov 11, 2022

Contents

CHAPTER 1

What is Meer

"Meer" is a dedicated data broker for the Suricata IDS/IPS system and the Sagan log analysis engine.

Meer takes EVE data (JSON) from Suricata or Sagan (via an `input-plugin`), augments it by enriching it with DNS, GeoIP, and other information (via the `meer-core`), and then pushes the data to a database (via a `output-plugin`) of your choice.

Meer is written in C which makes it fast and very light weight. This makes is suitable for processing data on systems with limited resource.

Meer `input-plugins` that are currently supported are Suricata/Sagan EVE ("spool") files and Redis.

Meer `output-plugins` that are currently supported are Elasticsearch, Opensearch, Zincsearch (https://github.com/zinclabs/zinc), Redis, named pipes, files, and "external" programs. Meer release 1.0.0 supports SQL (MariaDB, MySQL and PostgreSQL) that is compatible with older "Barnyard2" systems. Meer versions _after_ 1.0.0 do _not_ support SQL.

The primary Meer site is located at:

https://github.com/quadrantsec/meer

1.1 License

Meer is licensed under the GNU/GPL version 2.

CHAPTER 2

Installation

2.1 Quick start from source

Quick installation using common flags. For more information on packages and flags, skip to `Required Prerequisites` and `Optional Prerequisites`.

```
sudo apt-get install libjson-c-dev libyaml-dev libmaxminddb-dev libcurl4-openssl-dev libhiredis-dev libevent-dev zlib1g-dev
git clone https://github.com/quadrantsec/meer
cd meer
./autogen.sh
./configure --enable-redis --enable-elasticsearch --enable-geoip
make
sudo make install
```

By default, this will install Meer into the `/usr/local/bin/` directory with the default Meer configuration file in the `/usr/local/etc/` directory. By default (without any flags), Meer will compile with only Redis support.

2.2 Required Prerequisites

Meer uses a YAML configuration file. This means that Meer will need libyaml installed on the system. On Ubuntu/Debian systems, this can be installed via:

```
apt-get install libyaml-dev
```

Meer uses JSON-C to parse JSON (EVE) output from Sagan and Suricata. On Ubuntu/Debian systems, this prerequisite can be installed via:

```
apt-get install libjson-c-dev
```

2.3 Optional Prerequisites

2.3.1 Redis

If you would like to have Meer store data into Redis, which is enabled by default during compile time, you will need the "hiredis" library and development files. You will also need `libevent` installed as well.

On Ubuntu/Debian systems:

```
sudo apt-get install libhiredis-dev libevent-dev
```

2.3.2 Elasticsearch

If you would like Meer to use the 'elasticsearch' output plugin, then you'll need to install libcurl. To do this on Ubuntu/Debian systems, do the following:

```
apt-get install libcurl4-openssl-dev
```

2.3.3 Maxmind (GeoIP support)

If you would like Meer to add GeoIP data to Suricata/Sagan EVE data, you'll want to install the Maxmind (https://maxmind.com) library. To do this on Ubuntu/Debian systems, do the following:

```
apt-get install libmaxminddb-dev
```

2.3.4 JEMalloc

For JEMalloc support, on Debian/Ubuntu systems, install the JEMalloc library:

```
apt-get install libjemalloc-dev
```

2.3.5 TCMalloc

For TCMalloc support, on Debian/Ubuntu systems, install the TCMalloc library:

```
apt-get install libtcmalloc-minimal4
```

Common configure options

--prefix=/usr/
    Installs the Meer binary in the /usr/bin. The default is `/usr/local/bin`.

--sysconfdir=/etc
    Installs the Meer configuration file (meer.yaml) in the /etc directory. The default is `/usr/local/etc/`.

--with-libjsonc-libraries
    This option points Meer to where the json-c libraries reside.

--with-libjsonc-includes
    This option points Meer to where the json-c header files reside.

--with-libyaml_libraries
    This option points Meer to where the libyaml files reside.

--with-libyaml-includes
This option points Meer to where the libyaml header files reside.

--enable-redis
This option enables Redis output support. It requires "hiredis" to be installedt.

--enable-elastcisearch
This option enables Elastcisearch support. It requires "libcurl" to be installed.

--enable-geoip
This option enables Maxmind's GeoIP support. It requires "libmaxminddb" Maxmind library to be install.

--enable-bluedot
This optino allows Meer to write to a Bluedot "threat intel" database alert data via HTTP. This requres that "libcurl" be installed. You probably don't want this.

--enable-tcmalloc
This options enables support for Google's TCMalloc. For more information, see https://github.com/google/tcmalloc

--enable-jemalloc
This options enables support for JEMalloc. For more information, see https://jemalloc.net.

CHAPTER 3

Command Line Options

The majority of controls for Meer are within the `meer.yaml` file.

`-d, --daemon`
This option tells Meer to fork to the background.

`-c, --config`
This option tells what configuration file to use. By default Meer uses `/usr/local/etc/meer.yaml`.

`-h, --help`
The Meer help screen.

`-q, --quiet`
This option to tells Meer to not output to the console. Logs are still sent to the /var/log/meer directory.

`-q, --file`
This option bypasses the meer.yaml 'input-type' option and reads in files from the command line. Gzip compressed files can be read if Meer is compiled with GZIP support. If specifying multiple files, make sure to enclose your options with quotes (for example, –file "/var/log/suricata/*.gz")

CHAPTER 4

Starting Meer

To start Meer as root type:

```
/usr/local/bin/meer
```

To start Meer with a specified configuration file as root type:

```
/usr/local/bin/meer --config /path/to/my/config
```

To start Meer with a specified configuration file in “quiet” mode as root type:

```
/usr/local/bin/meer --config /path/to/my/config --quiet
```

to start Meer in the background as “root” type:

```
/usr/local/bin/meer --daemon
```

CHAPTER 5

Meer configuration:

Meers operations are mainly controlled by the `meer.yaml` file. The configuration file is split into three sections. The `meer-core` controls how Meer processes incoming data from EVE files. The `input-plugins` controls how Meer receives data. The `output-plugins` controls how data extracted from the EVE files is transported to a database backend. To view a full example `` `meer.yaml` `` configuration file, go to: https://github.com/quadrantsec/meer/blob/main/etc/meer.yaml

5.1 ‘core’ options

Below describes the options in the *core* section of the `meer.yaml`.

5.1.1 hostname

Texts field that is added to Suricata/Sagan EVE JSON. This short text field represents “were” the data is originating from. This is a required option. For example::

```
hostname: "awesome-sensor.example.com"
```

5.1.2 interface

This describes in what interface the data was collected. With Suricata, this might description the device network traffic is being acquired from (“etho”, “bridge0”, etc). With Sagan, this might describe log sources (“windows-logs”, “cisco-logs”, etc). This is a required option. For example::

```
interface: "eth0"
```

5.1.3 description

This is a text field that description the sensor (what it is monitoring, etc). This is typically a short sentence. For example::

```
description: "DMZ - web services and SQL databases".
```

This data is add to the Suricata or Sagan EVE data.

5.1.4 type

The `type` is a single text field to describe the sensor. At Quadrant Information Security, we use this field to describe the sensor function in life. For example::

```
type: "pie"           # PIE == Packet Inspection Engine / LAE == Log Analysis Engine
```

5.1.5 payload-buffer-size

The max memory to be allocate per EVE log line. This should match you Suricata or Sagan buffer size. If you EVE data is being truncated, consider increasing this. The default a `1mb of RAM::

```
payload-buffer-size: 1024kb  # Can end with kb, mb, gb.
```

5.1.6 runas

This is the user name the Meer process should "run as". You will likely want to run Meer as the same user name that is collecting information (for example, "suricata" or "sagan"). The `runas` can protect your system from security flaws in Meer. **Do not run as "root"**. This option is required::

```
runas: "suricata"
```

5.1.7 classification

The `classification` option tells Meer where to find classification types. This file typically ships with Sagan, Suricata, and Snort rules. It defines a 'classtype' (for example, "attempt-recon") and assigns a numeric priority to the event. This option is required::

```
classification: "/etc/suricata/classification.config"
```

5.1.8 meer_log

The `meer_log` is the location of the file for Meer to record errors and statistics to. The file will need to be writable by the same user specified in the `runas` option. If not specified, the default file location is `/var/log/meer.log`.::

```
meer_log: "/var/log/meer/meer.log"
```

5.1.9 lock_file

The lock_file is used to help avoid multiple Meer processes from processing the same data. The lock_file should be unique per Meer instance. The lock file contains the process ID (PID) of instance of Meer. This option is required.::

```
lock_file: "/var/log/meer/meer.lck"
```

5.1.10 input-type

This tells Meer where to acquire data from. This controls which input plugin (input-plugins) to use. This option is required.::

```
input-type: "file"
```

5.1.11 calculate-stats

When statistics (event_type "stats") from Suricata are collected, they are represented in a accumulated manor (ie - "1000,2000,3000,4000"). While this works well for some utilities (rrdtool , librenms, etc), it doesn't work well with others (SQL databases, etc). When this option is enabled, Meer will track and do the math to convert the statistics as a accumulated metric (ie "1000, 2000, 3000, 4000") to time based, between "stats" metric (ie - "1000,1000,1000,1000"). Another example would be, rather than reporting Suricata has seen X number of bytes since this initial start of Suricata, X number of bytes has been seen since the last statistics where reported. This option does not process all stats but rather a small subset. They are kernel_packets, kernel_drops, errors, bytes, invalid, ipv4, ipv6, tcp and udp. When the calculate-stats option is enabled, a new JSON nest is added to the event_type stats with these aggregate statistics. ::

```
calculate-stats: false
```

5.1.12 fingerprint

The fingerprint option tells Meer to decode "fingerprint" rules and route the data differently. Fingerprint rules do not work like normal rules. The data from these rules is used to passively fingerprint systems for operating systems and types (client/server). This information can be valuable to determine if an attack might have been successful or not.

For a full explanation of our Meer handles Suricata and Sagan "fingerprinting" signatures, please watch Jeremy Groves "Passive Fingerprinting Suricata" on Youtube (https://www.youtube.com/watch?v=n5O4-iqAlVo). ::

```
fingerprint: disabled
fingerprint_networks: "10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12"

fingerprint_reader: enabled         # This option appends "fingerprint"
                                    # data to "alert".

fingerprint_writer: enabled         # This option detects "fingerprint"
                                    # alerts and writes them to Redis.
```

The fingerprint_networks are you networks. These are the IP address spaces we want to record device fingerprint data from. The fingerprint_reader tells Meer to "append" fingerprint data to alert EVE JSON. The ` fingerprint_writer configures Meer to "write" fingerprint data about devices to Redis. By default, this option is disabled.

5.1.13 client_stats

This option has no affect on Suricata data. This option can be used when processing Sagan data. The `client_stats` option works in conjunction with the Sagan `client-stats` option. The basic concept is that Sagan will write out information at intervals (example log data, bytes sent from individual clients, etc). This option will read in this JSON and report it to a Redis backend. By default, this option is disabled.::

```
client_stats: disabled
```

5.1.14 oui_lookup

When Meer encounters a MAC address within an EVE file, it will lookup the vendor of the MAC address. This data is added to the EVE JSON. By default, this is disabled. ::

```
oui_lookup: disabled
oui_filename: "/usr/local/etc/manuf"
                                           # https://gitlab.com/wireshark/wireshark/
↪raw/master/manuf
                                           # This file contains MAC/OUI data.
```

5.1.15 dns

The `dns` option tells Meer to perform a DNS PTR (reverse) record lookup of the IP addresses involved in an alert. This option is useful because it records the DNS in your EVE JSON at the time the event occurred. This is enabled by default. ::

```
dns: enabled
dns_cache: 900                             # Time in seconds / cache timeout
dns_lookup_types: "alert,ssh,http,rdp,ftp"  # The event_type to do DNS
                                           # PTR lookups for.  This can
                                           # be the event_type or "all".
```

When `dns` is enabled, Meer will internally cache records to avoid repetitive lookups. For example, if 1000 alerts come in from a single IP address, Meer will look up the DNS PTR record one time and use the cache for the other 999 times. This saves on lookup time and extra stress on the internal DNS server. If you do not want Meer to cache DNS data, simply set this option to 0. The `dns_cache` time is in seconds.

`dns_lookup_types` are Suricata `event_types` that DNS queries will be performed on.

5.1.16 geoip

If Meer is compiled with the `--enable-geoip` option, this will allow Meer to do GeoIP lookups from a Maxmind (https://maxmind.com) data. GeoIP information is stored within the EVE JSON as a new JSON nest named `geoip_src` and `geoip_dest`. This data can include country code, subdivision, City, postal code, timezone, longitude and longitude. By default, this option is disabled. ::

```
geoip: disabled
geoip_database: "/usr/local/share/GeoIP2/GeoLite2-City.mmdb"
```

The `geoip_database` is the location of your Maxmind database file. This is loaded when Meer is started. You can download GeoIP "Lite" databases from https://dev.maxmind.com/geoip/geolite2-free-geolocation-data

5.1.17 ndp-collector

The NDP collector (Network Data Point) is an option of distilling data from Suricata into “non-repetitive” data points. The concept is that store data into Elasticsearch, Opensearch or Zincsearch (https://github.com/zinclabs/zinc) for “quick” IOC (Indicator of Compromise) searches. Since the data is “non-repetitive”, the NDP collector only stores the minimal amount of data around an event. This option is disabled by default. We will be adding more information about this option as it comes available. ::

```
ndp-collector: disabled
ndp-debug: disabled
ndp-ignore-networks: "10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12"
ndp-routing: "flow, http, ssh, fileinfo, tls, dns, smb, ftp"

ndp-smb: "SMB2_COMMAND_CREATE, SMB2_COMMAND_WRITE"
ndp-smb-internal: true

ndp-ftp: "STOR, RETR, USER"
```

The `ndp-ignore-networks` should represent any public or internal network blocks you use. The NDP collector not store data about these networks as they are typically not useful for rapid IoC searches.

The `ndp-routing` tells Meer where to pull non-repetitive data from. Since we are storing non-repetitive data, the only options are flow, http, ssh, fileinfo, tls, dns, smb and ftp.

The `ndp-smb` option configures Meer to only store SMB command related to this list. Typically, to keep datasets small, we only want to record SMB2_COMMAND_CREATE and SMB2_COMMAND_WRITE. Because SMB is not typically used over the Internet, the `ndp-smb-internal` option configures Meer to record all internal SMB traffic. This is done because SMB is used by attackers to move laterally within a network.

The `ndp-ftp` option records FTP traffic but only commands related to this list.

If this option is being used, use the `input-type` of `redis` is probably the most efficient.

CHAPTER 6

Output Plugins

6.1 redis

This controls how Meer logs to a Redis database. Meer can record alert records to Redis similar to how Suricata with Redis support enabled does. Redis is also used as a temporary storage engine for `client_stats` (Sagan only) and `fingerprint` data if enabled.

```
###########################################################################
# redis
#
# This allows you to send Suricata/Sagan EVE data to a Redis database.
# This will mimic the way Suricata writes EVE data to Redis with the
# exception of "client_stats" which is a Sagan specific processor.
###########################################################################

redis:

  enabled: no
  debug: no
  server: 127.0.0.1
  #password: "mypassword"
  port: 6379
  batch: 1                   # Batching (pipelining) data.  When set to 1,
                             # no batching is performed and data is immediately
                             # sent to Redis.  If increase,  data is batched
                             # and sent in bulk to increase performance.  The max
                             # is 100.
  key: "suricata"            # Default 'channel' to use.  If none is specified, the
                             # channel name will become the "event_type".
                             # (ie - alert, dhcp, dns, flow, etc).
  mode: lpush                # How to publish data to Redis.  Valid types are
                             # "list" ("lpush"), "rpush", "channel" ("publish"),
                             # "set".
  append_id: disabled        # If enabled, this will append the "hostname" and
```

(continues on next page)

(continued from previous page)

```
                           # waldo position to the key.  For example,  the
                           # Redis object can become "alert|hostname|1". This
                           # is good when you are using the "set" mode.

  routing:

    - alert
    - files
    - flow
    - dns
    - http
    - tls
    - ssh
    - smtp
    - email
    - fileinfo
    - dhcp
    - stats
    - rdp
    - sip
    - ftp
    - ikev2
    - nfs
    - tftp
    - smb
    - dcerpc
    - mqtt
    - netflow
    - metadata
    - dnp3
    - anomaly
    - fingerprint

  # This controls sending Sagan client tracking data to Redis.  This has no affect
  # on Suricata systems.

    - client_stats
```

The mode controls how data is stored to Redis. Valid options are list, lpush, rpush, channel or publish. The default is list. The method Meer stores the data is compatible with Suricata's Redis output format. Note; This option does not have any affect on client_stats or fingerprint recording.

The routing option tells Meer "what" Suricata or Sagan to store in Redis.

6.2 elasticsearch

This option enables the Elasticsearch output. This option is compatible with Opensearch, Elasticsearch and Zincsearch (https://github.com/zinclabs/zinc).

```
############################################################################
# elasticsearch
#
# This section allows you to route data to Elasticsearch.  This module
# supports authentication and TLS support.
```

(continues on next page)

(continued from previous page)

```
##########################################################################

elasticsearch:

  enabled: no
  debug: no
  url: "http://127.0.0.1:9200/_bulk"
  index: "suricata_$EVENTTYPE_$YEAR$MONTH$DAY"
  insecure: true                                      # Only applied when https is␣
↪used.
  batch: 100                                          # Batch size per/writes.
  threads: 10                                         # Number of "writer" threads.
  #username: "myusername"
  #password: "mypassword"

  routing:

    - alert
    - files
    - flow
    - dns
    - http
    - tls
    - ssh
    - smtp
    - email
    - fileinfo
    - dhcp
    - stats
    - rdp
    - sip
    - ftp
    - ikev2
    - nfs
    - tftp
    - smb
    - dcerpc
    - mqtt
    - netflow
    - metadata
    - dnp3
    - anomaly
    - fingerprint
    - ndp
```

6.3 external

This option allows signatures to call "external" programs. For example, if a signature the proper "metadata" (`metadata: meer external` or a set policy), Meer will fork a copy of the specified program and pass the EVE via stdin. This feature can be useful for creating custom firewalling routines or routing data to alternate programs. The "external" program can be written in any language that suites you.

```
##########################################################################
# external
```

(continues on next page)

(continued from previous page)

```
#
# EVE data (JSON) is passed via stdin to the external program.   The
# external program can be written in any language you choose (shell script,
# Python, Perl, etc).
#
# This can be useful for automatic firewalling,  building block lists,
# replicating "snortsam" functionality, etc.  See the "tools/external"
# directory for example routines that use this feature.
#
# If this option is enabled, any rule that has the metadata of "meer
# external" (ie - "metadata:meer external") will cause the external script
# to be executed.  Execution can also be controlled by Snort metadata
# "policies".
##########################################################################

external:

  enabled: no
  debug: no

  # Execution of an external program based on metadata "policy".  When Meer
  # encounters a "policy" (security-ips, balanced-ips, connectivity-ips,
  # and max-detect-ips),  Meer will execute the specified routine.
  # Currently only Snort rules have these types of polices.  This can be
  # useful when you want to execute an external script that will to "block"
  # or "firewall" based off the policy types.  This section only applies if
  # you are using Suricata with Snort rules.  Snort's polices are
  # below:

  # connectivity-ips  - You run a lot of real time applications (VOIP,
  # financial transactions, etc), and don't want to run any rules that
  # could affect the current performance of your sensor.  The rules in this
  # category make snort happy, additionally this category focuses on the high
  # profile most likely to affect the largest number of people type of
  # vulnerabilities.

  # balanced-ips - You are normal, you run normal stuff and you want normal
  # security protections.  This is the best policy to start from if you are
  # new, old, or just plain average.  If you don't have any special
  # requirements for super high speeds or super secure networks start here.

  # security-ips - You don't care about dropping your bosses email, everything
  # in your environment is tightly regulated and you don't tolerate people
  # stepping outside of your security policy.  This policy hates on IM, P2P,
  # vulnerabilities, malware, web apps that cause productivity loss, remote
  # access, and just about anything not related to getting work done.
  # If you run your network with an iron fist start here.

  # I can't seem to find any documentation on what "max-detect-ips" is :(

  program: "/usr/local/bin/external_program"

  #meer_metadata: enabled
  #cisco_policies: "policy-security-ips,policy-max-detect-ips,policy-connectivity-ips,
↪policy-balanced-ips"
  #et_signature_severity: "critical,major"            # Critical,Major,Minor,
↪Informational
```

(continues on next page)

(continued from previous page)

```
  # You likely don't want to route to much data to a external program. External
  # output is slow.

  routing:

    - alert
```

6.4 pipe

Below is an example of the "pipe" output plugin. This takes data being written to the EVE file and puts it into a named pipe (FIFO). This can be useful if you want a third party program (for example, Sagan - https://sagan.io) to analyze the data.

```
##########################################################################
# pipe
#
# This allows Meer to send a copy of an event to a named pipe (FIFO) in
# its raw,  JSON form.  This allows for third party tools, like Sagan,
# to do further analysis on the event.
##########################################################################

pipe:

  enabled: no
  pipe_location: /var/sagan/fifo/sagan.fifo
  pipe_size: 1048576                        # System must support F_GETPIPE_SZ/F_
↪SETPIPE_SZ

  routing:

    - alert
    - files
    - flow
    - dns
    - http
    - tls
    - ssh
    - smtp
    - email
    - fileinfo
    - dhcp
    - stats
    - rdp
    - sip
    - ftp
    - ikev2
    - nfs
    - tftp
    - smb
    - dcerpc
    - mqtt
    - netflow
    - metadata
```

(continues on next page)

(continued from previous page)

```
    - dnp3
    - anomaly
    - fingerprint
```

CHAPTER 7

Console Output

7.1 Console/Log Startup

At start up, the logs and console output give you information about the status of Meer. For example, you will want to note the Redis and Elasticsearch, such as the driver and whether a successful connection was made. If there is a problem making a connection to your database, Meer will display the error that is causing the issues.

Another important item to note is the database sensor ID. This will be the ID number used in the database to store events.

Common issues are database rights and directory/file permission problems.

If Meer makes it to the Waiting of new data..., then Meer has successfully started.

```
[*] [10/20/2021 20:55:23] Configuration '/usr/local/etc/meer.yaml' for host 'dev'␣
↪successfully loaded.
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23]  @@@@@@@@@@  @@@@@@@@ @@@@@@@@ @@@@@@@    Meer version 1.0.
↪0-git
[*] [10/20/2021 20:55:23]  @@! @@! @@! @@!      @@!      @@!  @@@   Quadrant␣
↪Information Security
[*] [10/20/2021 20:55:23]  @!! !!@ @!@ @!!!:!   @!!!:!   @!@!!@a    https://
↪quadrantsec.com
[*] [10/20/2021 20:55:23]  !!:     !!: !!:      !!:      !!: :!a    Copyright (C)␣
↪2018-2021
[*] [10/20/2021 20:55:23]   :      :   : :: ::  : :: ::   :   : :
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] Meer's PID is 14606
[*] [10/20/2021 20:55:23] Dropping privileges! [UID: 1011 GID: 1011]
[*] [10/20/2021 20:55:23] Loaded 40382 entries from OUI database [/usr/local/etc/
↪manuf].
[*] [10/20/2021 20:55:23] Classifications file loaded [/usr/local/etc/sagan-rules/
↪classification.config].
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] Fingerprint support    : enabled
```

(continues on next page)

(continued from previous page)

```
[*] [10/20/2021 20:55:23] Health updates         : enabled
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] GeoIP support          : enabled
[*] [10/20/2021 20:55:23] GeoIP database         : /usr/local/share/GeoIP2/GeoLite2-
↪City.mmdb
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] Waldo loaded. Current position: 2345
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] --[ Redis output information ]------------------------------
↪--------
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] Successfully connected to Redis server at 127.0.0.1:6379.
[*] [10/20/2021 20:55:23] Got PONG from Redis at 127.0.0.1:6379.
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] Write 'alert'        : enabled
[*] [10/20/2021 20:55:23] Write 'stats'        : enabled
[*] [10/20/2021 20:55:23] Write 'email'        : enabled
[*] [10/20/2021 20:55:23] Write 'dns'          : enabled
[*] [10/20/2021 20:55:23] Write 'flow'         : enabled
[*] [10/20/2021 20:55:23] Write 'http'         : enabled
[*] [10/20/2021 20:55:23] Write 'tls'          : enabled
[*] [10/20/2021 20:55:23] Write 'ssh'          : enabled
[*] [10/20/2021 20:55:23] Write 'smtp'         : enabled
[*] [10/20/2021 20:55:23] Write 'files'        : enabled
[*] [10/20/2021 20:55:23] Write 'fileinfo'     : enabled
[*] [10/20/2021 20:55:23] Write 'dhcp'         : enabled
[*] [10/20/2021 20:55:23] Write 'rdp'          : enabled
[*] [10/20/2021 20:55:23] Write 'sip'          : enabled
[*] [10/20/2021 20:55:23] Write 'ftp'          : enabled
[*] [10/20/2021 20:55:23] Write 'ikev2'        : enabled
[*] [10/20/2021 20:55:23] Write 'nfs'          : enabled
[*] [10/20/2021 20:55:23] Write 'tftp'         : enabled
[*] [10/20/2021 20:55:23] Write 'smb'          : enabled
[*] [10/20/2021 20:55:23] Write 'dcerpc'       : enabled
[*] [10/20/2021 20:55:23] Write 'mqtt'         : enabled
[*] [10/20/2021 20:55:23] Write 'netflow'      : enabled
[*] [10/20/2021 20:55:23] Write 'metadata'     : enabled
[*] [10/20/2021 20:55:23] Write 'dnp3'         : enabled
[*] [10/20/2021 20:55:23] Write 'anomaly'      : enabled
[*] [10/20/2021 20:55:23] Write 'client_stats' : enabled
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] --[ Elasticsearch output information ]----------------------
↪-----
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] URL to connect to       : "https://127.0.0.1:9200/_bulk"
[*] [10/20/2021 20:55:23] Index template          : "suricata_$EVENTTYPE_$YEAR$MONTH
↪$DAY"
[*] [10/20/2021 20:55:23] Batch size per/POST     : 100
[*] [10/20/2021 20:55:23] Threads                 : 10
[*] [10/20/2021 20:55:23] Authentication          : enabled
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] Record 'alert'    : enabled
[*] [10/20/2021 20:55:23] Record 'files'    : enabled
[*] [10/20/2021 20:55:23] Record 'flow'     : enabled
[*] [10/20/2021 20:55:23] Record 'dns'      : enabled
[*] [10/20/2021 20:55:23] Record 'http'     : enabled
[*] [10/20/2021 20:55:23] Record 'tls'      : enabled
```

(continues on next page)

(continued from previous page)

```
[*] [10/20/2021 20:55:23] Record 'ssh'      : enabled
[*] [10/20/2021 20:55:23] Record 'smtp'     : enabled
[*] [10/20/2021 20:55:23] Record 'email'    : enabled
[*] [10/20/2021 20:55:23] Record 'fileinfo' : enabled
[*] [10/20/2021 20:55:23] Record 'dhcp'     : enabled
[*] [10/20/2021 20:55:23] Record 'stats'    : enabled
[*] [10/20/2021 20:55:23] Record 'rdp'      : enabled
[*] [10/20/2021 20:55:23] Record 'sip'      : enabled
[*] [10/20/2021 20:55:23] Record 'ftp'      : enabled
[*] [10/20/2021 20:55:23] Record 'nfs'      : enabled
[*] [10/20/2021 20:55:23] Record 'tftp'     : enabled
[*] [10/20/2021 20:55:23] Record 'smb'      : enabled
[*] [10/20/2021 20:55:23] Record 'mqtt'     : enabled
[*] [10/20/2021 20:55:23] Record 'dcerpc'   : enabled
[*] [10/20/2021 20:55:23] Record 'netflow'  : enabled
[*] [10/20/2021 20:55:23] Record 'metadata' : enabled
[*] [10/20/2021 20:55:23] Record 'dnp3'     : enabled
[*] [10/20/2021 20:55:23] Record 'anomaly'  : enabled
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] Spawning 10 Elasticsearch threads.
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] --[ Meer engine information ]-------------------------------
↪------
[*] [10/20/2021 20:55:23]
[*] [10/20/2021 20:55:23] Successfully opened /home/champ/test.eve
[*] [10/20/2021 20:55:23] Skipping to record 2345 in /home/champ/test.eve
[*] [10/20/2021 20:55:23] Reached target record of 2345.  Processing new records.
[*] [10/20/2021 20:55:23] Read in 2345 lines
[*] [10/20/2021 20:55:23] Waiting for new data......
```

CHAPTER 8

Getting help

The Meer Github site is located at:

https://github.com/quadrantsec/meer

If you are having issues getting Meer to work, consider posting in the Meer mailing list. This list is good for general configuration, install, and usage questions.

https://groups.google.com/forum/#!forum/meer-users

If you need to report a compile or programming issue, please use our Github.com issues page. That is located at:

https://github.com/quadrantsec/meer/issues

You can also get support via our Meer Discord channel. That is at:

https://discord.gg/VS6jTjH4gW

Index