Meer User Guide Documentation

Release 0.0.3

Champ Clark III

Jan 04, 2022

Contents

1 What is Meer **1**
- 1.1 License . . . 1

2 Installation **3**
- 2.1 Required Prerequisites . . . 3
- 2.2 Optional Prerequisites . . . 3
- 2.3 Source . . . 4

3 Command Line Options **5**

4 Starting Meer **7**

5 'core' configuration: **9**
- 5.1 'meer-core' example . . . 9
- 5.2 'meer-core' options . . . 11

6 Output Plugins **15**
- 6.1 Redis . . . 15
- 6.2 Elasticsearch . . . 17
- 6.3 External . . . 18
- 6.4 Pipe . . . 20
- 6.5 File . . . 21

7 Console Output **23**
- 7.1 Console/Log Startup . . . 23

8 Getting help **27**

Index **29**

CHAPTER 1

What is Meer

"Meer" is a dedicated "spooler" for the Suricata IDS/IPS and Sagan log analysis engines. This means that as Suricata or Sagan write alerts out to a file, Meer can 'follow' that file and store the alert information into a database. You can think of the "spool" file as a 'queuing' system for alerts from Suricata or Sagan. Using a "spooling" system ensures the delivery of alerts to a back end database. This task was traditionally accomplished by using a file format called "unified2" which was developed by the SourceFire/Snort team and a program called Barnyard2. While unified2 has been useful, its binary nature makes it difficult to work with and it has not been extended in quite some time. Instead of following "unified2" files, Meer follows Suricata and Sagan's "EVE" (JSON) output format. Since the EVE output format is JSON, it is easier to work with. The EVE output also contains valuable information that does not exist in "unified2".

Meer is meant to be modular and simple. This project does not aim to replicate all features of Barnyard2. The idea is to replicate the more useful features and abandon the "cruft".

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

There are currently no binary packages of Meer available. However, installation from source is pretty straightforward.

2.1 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
```

2.2 Optional Prerequisites

2.2.1 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. On Ubuntu/Debian systems:

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

2.2.2 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 Source

Installation from source distributions files gives

Basic steps:

```
git clone https://github.com/quadrantsec/meer
cd meer
./autogen.sh
./configure
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, Meer will compile with only Redis support.

--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

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.

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

'core' configuration:

Meers operations are mainly controlled by the `meer.yaml` file. The configuration file is split into two sections. The `meer-core` controls how Meer processes incoming data from EVE files. The `output-plugins` controls how data extracted from the EVE files is transported to a database backend.

5.1 'meer-core' example

```
meer-core:

core:

  hostname: "mysensor"  # Unique name for this sensor (no spaces)
  interface: "eth0"     # Can be anything.  Sagan "syslog", suricata "eth0".

  description: "My awesome sensor!"   # Description of this sensor.  This
                                      # will be added to _all_ logs!

  payload-buffer-size: 1024kb         # This is the max size buffer that can
                                      # be read in/written out.  It should
                                      # match your payload-buffer-size in
                                      # Surcata or be larger.  Valid
                                      # Notations are "kb", "mb" and "gb".

  runas: "suricata"     # User to "drop privileges" too.
  #runas: "sagan"

  classification: "/etc/suricata/classification.config"
  #classification: "/usr/local/etc/sagan-rules/classification.config"

  meer_log: "/var/log/meer/meer.log"          # Meer log file
  waldo_file: "/var/log/meer/meer.waldo"      # Where to store the last
                                              # position in the
                                              # "follow-eve" file.
```

(continues on next page)

(continued from previous page)

```
 lock_file: "/var/log/meer/meer.lck"         # To prevent dueling processes.

 follow_eve: "/var/log/suricata/alert.json"  # The Suricata/Sagan file to monitor
 #follow-eve: "/var/log/sagan/alert.json"

##############################################################################
 # fingerprint
 #
 # This enables the "fingerprint" option.  When used in conjunction with the
 # "fingerprint.rules" (https://github.com/quadrantsec/fingerprint-rules),
 # this will record things like operating system type,  type of system it is
 # (client/server), etc.  This data get routed differently and does not
 # generate "alerts".
 ##############################################################################

 fingerprint: enabled
 fingerprint_networks: "10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12"
 ##############################################################################
 # client_stats
 #
 # "client_stats" are specific to Sagan and allow Sagan/Meer to record
 # information about systems sending Sagan data.  This has no affect on
 # Suricata.
 ##############################################################################

 client_stats: disabled

 ##############################################################################
 # oui_lookup
 #
 # The "oui_lookup" allows Meer to lookup vender information based off
 # a MAC address.  Information is stored in fingerprinting JSON.  The
 # MAC/OUI database
 #
 # https://gitlab.com/wireshark/wireshark/raw/master/manuf
 ##############################################################################

 oui_lookup: disabled
 oui_filename: "/usr/local/etc/manuf"

##############################################################################
 # dns
 #
 # If "dns" is enabled, Meer will do reverse DNS (PTR) lookups of an IP.
 # The "dns_cache" is the amount of time Meer should "cache" a PTR record
 # for.  The DNS cache prevents Meer from doing repeated lookups of an
 # already looked up PTR record.  This reduces overloading DNS servers.
 ##############################################################################

 dns: enabled
 dns_cache: 900      # Time in seconds.

 ##############################################################################
 # geoip
 #
 # If "geoip" is enabled, Meer will add GeoIP information (JSON) to
```

(continues on next page)

(continued from previous page)

```
# "alert" data.  You'll need to compile Meer with Maxmind's GeoIP
# support (--enable-geoip). Data that will be added,  when available,
# includes ISO country code, city, subdivision, postal code,
# timezone,  latitude and longitude.
#####################################################################

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

5.2 'meer-core' options

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

5.2.1 hostname

This is stored in the database in the `sensor` table under the `hostname` column. The `interface` is appended to the `hostname`. This option is required.

5.2.2 interface

The `interface` is stored in the `sensor` table appended to the `hostname` and `interface` columns. This describes in what interface the data was collected. This can be any descriptive string. For example, "eth0", "syslog", etc. This option is required.

5.2.3 runas

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

5.2.4 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.

5.2.5 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.

5.2.6 metadata

The `metadata` option tells Meer to decode "metadata" from Suricata or Sagan. If the "metadata" is present in the alert, Meer will decode it and store its contents in memory for later use.

5.2.7 flow

The `flow` option tells Meer to decode "flow" data from Suricata or Sagan. If the "flow" JSON is present in the alert, Meer will decode it and store its contents in memory for later use.

5.2.8 http

The `http` option tells Meer to decode "http" data from Suricata or Sagan. If the "http" JSON is present in the alert, Meer will decode it and store its contents in memory for later use.

5.2.9 tls

The `tls` option tells Meer to decode "tls" data from Suricata or Sagan. If the "tls" JSON is present in the alert, Meer will decode it and store its contents in memory for later use.

5.2.10 ssh

The `ssh` option tells Meer to decode "ssh" data from Suricata or Sagan. If the "ssh" JSON is present in the alert, Meer will decode it and store its contents in memory for later use.

5.2.11 smtp

The `smtp` option tells Meer to decode "smtp" data from Suricata or Sagan. If the "smtp" JSON is present in the alert, Meer will decode it and store its contents in memory for later use.

5.2.12 email

The `email` option tells Meer to decode "email" data from Suricata or Sagan. If the "email" JSON is present in the alert, Meer will decode it and store its contents in memory for later use. This is not to be confused with `smtp`. The data from `email` will contain information like e-mail file attachments, carbon copies, etc.

5.2.13 json

The `json` option tells Meer to store the original JSON/EVE event. This is the raw event that Meer has read in.

5.2.14 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. Fingerprint rules are located at https://github.com/quadrantsec/fingerprint-rules.

5.2.15 fingerprint_log

When fingerprint rules fire, this is the log file that is create and data sent to. This log file format is an JSON (EVE) log file and is meant to be routed to a Elasticsearch back end. The idea is to store this information for historical purposes.

5.2.16 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 record at the time the event occurred.

5.2.17 dns_cache

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.

5.2.18 health

The `health` option is a set of signatures used to monitor the health of Meer and your Sagan or Suricata instances. When enabled, Meer will treat certain Sagan and Suricata signatures as "health" indicators rather than normal alerts. When a "health" signature occurs, Meer updates the `sensor` table `health` column with the epoch time the health signature triggered. This can be useful in quickly determining if a sensor is down or behind (back logged) on alerts.

5.2.19 health_signatures

When `health` is enabled, this option supplies a list of signature IDs (sid) to Meer of Suricata or Sagan "health" signatures.

5.2.20 waldo_file

The `waldo_file` is a file that Meer uses to keep track of its last location within a EVE/JSON file. This keeps Meer from re-reading data in between stop/starts. This option is required.

5.2.21 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.

5.2.22 follow_eve

The `follow_eve` option informs Meer what file to "follow" or "monitor" for new alerts. You will want to point this to your Sagan or Suricata "alert" EVE output file. You can think of Meer "monitoring" this file similar to how "tail -f" operates. This option is required.

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
```

6.1.1 enabled

Enable or disable the Redis output.

6.1.2 debug

Enable or disabled Redis debugging.

6.1.3 server

The Redis server address you want to store data to.

6.1.4 port

Port of the target Redis server.

6.1.5 batch

The `batch` is the amount of data to collect before sending it to Redis. This has no affect when using Redis with either `client_stats` or `fingerprint` data.

6.1.6 key

The `key` is the default Redis channel or key to use.

6.1.7 mode

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.

6.2 Elasticsearch

This option enables the Elasticseaerch output.

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

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
```

(continues on next page)

(continued from previous page)

```
    - ssh
    - smtp
    - email
    - fileinfo
    - dhcp
    - stats
    - rdp
    - sip
    - ftp
    - ikev2
    - nfs
    - tftp
    - smb
    - dcerpc
    - mqtt
    - netflow
    - metadata
    - dnp3
    - anomaly
    - fingerprint
```

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
#
# 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
```

(continues on next page)

(continued from previous page)

```
  # 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

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

  routing:

    - alert
```

6.3.1 enabled

Keyword is used to enable/disable `external` output.

6.3.2 debug

When enabled, this option will display and log debugging information.

6.3.3 policy-security-ips

Execute `external` program when the `policy-security-ips` is encountered.

6.3.4 policy-max-detect-ips

Execute `external` program when the `policy-max-detect-ips` is encountered.

6.3.5 policy-connectivity-ips

Execute `external` program when the `policy-connectivity-ips` is encountered.

6.3.6 policy-balanced-ips

Execute `external` program when the `policy-balanced-ips` is encountered.

6.3.7 program

`external` program to execute when conditions are met.

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
```

(continues on next page)

(continued from previous page)

```
    - rdp
    - sip
    - ftp
    - ikev2
    - nfs
    - tftp
    - smb
    - dcerpc
    - mqtt
    - netflow
    - metadata
    - dnp3
    - anomaly
    - fingerprint
```

6.4.1 enabled

Enabled/disabled the 'pipe' output.

6.4.2 pipe_location

Location of the named pipe on the file system.

6.4.3 pipe_size

Number of bytes will set the size of the named pipe/FIFO to.

6.5 File

This configures the 'file' output plugin.

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

Symbols

-enable-bluedot
    command line option, 4
-enable-elastcisearch
    command line option, 4
-enable-geoip
    command line option, 4
-enable-redis
    command line option, 4
-enable-tcmalloc
    command line option, 4
-prefix=/usr/
    command line option, 4
-sysconfdir=/etc
    command line option, 4
-with-libjsonc-includes
    command line option, 4
-with-libjsonc-libraries
    command line option, 4
-with-libyaml-includes
    command line option, 4
-with-libyaml_libraries
    command line option, 4
-c, -config
    command line option, 5
-d, -daemon
    command line option, 5
-h, -help
    command line option, 5
-q, -quiet
    command line option, 5

A

apt-get install libcurl4-openssl-dev
    command line option, 3
apt-get install libjson-c-dev
    command line option, 3
apt-get install libyaml-dev
    command line option, 3

C

command line option
    -enable-bluedot, 4
    -enable-elastcisearch, 4
    -enable-geoip, 4
    -enable-redis, 4
    -enable-tcmalloc, 4
    -prefix=/usr/, 4
    -sysconfdir=/etc, 4
    -with-libjsonc-includes, 4
    -with-libjsonc-libraries, 4
    -with-libyaml-includes, 4
    -with-libyaml_libraries, 4
    -c, -config, 5
    -d, -daemon, 5
    -h, -help, 5
    -q, -quiet, 5
    apt-get install
        libcurl4-openssl-dev, 3
    apt-get install libjson-c-dev, 3
    apt-get install libyaml-dev, 3
    sudo apt-get install
        libhiredis-dev, 3

S

sudo apt-get install libhiredis-dev
    command line option, 3