EveBox Documentation

Jason Ish

Mar 27, 2020

Contents:

CHAPTER 1

Installation

EveBox can be installed in a few ways:

- Simply unpacking a zip and running the EveBox binary.
- Installing an RPM.
- Installing a Debian package.

There are also RPM and Debian package repositories for installing with `yum/dnf` or `apt-get`.

To download go to the *Download* section of the EveBox web site.

Other installation instructions exist on the GitHub wiki.

CHAPTER 2

Server

2.1 Running

2.1.1 Using an Existing ELK Stack

Assuming you already have an existing working Suricata, Elastic Search, Logstash and Kibana stack working, then EveBox should just work if pointed at your Elastic Search server.

Example:

```
evebox -v -e http://elasticsearch:9200
```

This assumes the use of the default Logstash index logstash-{YYYY.MM.DD}. If another index name is being used it must be specified with the `-i` option:

```
evebox -v -e http://elasticsearch:9200 -i indexprefix
```

2.1.2 Consuming Events and Using Elastic Search

If you do not have an existing ELK stack, but are able to provide Elastic Search, EveBox can ship the events to Elastic Search itself.

Example usage:

```
evebox -v -e http://elasticsearch:9200 --input /var/log/suricata/eve.json
```

Note: If you do not wish to run EveBox on the same machine as Suricata you can use the *EveBox Agent* to ship alerts to the EveBox server.

2.1.3 Using the Embedded SQLite Database

If installing Elastic Search is not an option the embedded SQLite database can be used instead:

```
evebox -v -D . --datastore sqlite --input /var/log/suricata/eve.json
```

Note: Note the -D parameter that tells EveBox where to store data files such as the file for the SQLite database. While using the current directory, or a temp directory is OK for testing, you may want to use something like /var/lib/evebox for long term use.

2.2 Oneshot Mode

Oneshot mode is the running of EveBox to look at an eve.json a single file. On exit, everything is cleaned up.

Example:

```
evebox oneshot /var/log/suricata/eve.json
```

After reading all the events the browser should open to the *Inbox* display.

2.3 Authentication

2.3.1 Enabling Authentication

Authentication requires:

- Enabling authentication in your EveBox configuration file:

  ```
  authentication:
    required: true
    type: usernamepassword
  ```

- And enabling the configuration database either with the `-D` command line option or the `data-directory` configuration file setting.

Note: If using the RPM or Debian packages AND starting EveBox with systemd, the data-directory is already configured to be `/var/lib/evebox`.

Note: For the rest of this documentation, `/var/lib/evebox` will be used as the data-directory.

Starting the Server with Authentication Enabled

If EveBox was installed with a RPM or Debian package and started with *systemd*, it is already setup with a configuration database, just enable authentication in `/etc/evebox/evebox.yaml` (you may first need to `cp /etc/evebox/evebox.yaml.example /etc/evebox/evebox.yaml`.

Otherwise, if you are manually starting EveBox you must use the `-D` command line option to set the data directory where the configuration database can be stored:

```
./evebox server -D ~/.evebox/
```

Note: The `EVEBOX_DATA_DIRECTORY` environment variable can also be used to set the data directory.

2.3.2 Adding a User

Adding users is done with the config tool, for example:

```
evebox config -D /var/lib/evebox users add --username joe
```

Note: RPM and Debian package installations of EveBox setup */var/lib/evebox* to be owned by the user `evebox`, so you may need use sudo to add users, for example:

```
sudo -u evebox config -D /var/lib/evebox users add
```

or as root:

```
sudo evebox config -D /var/lib/evebox users add
```

2.3.3 External Authenticators

GitHub Oauth2

EveBox can authenticate against GitHub using Oauth2. It is required that the user is first created in the configuration database, but with the `--github-username` parameters, for example:

```
evebox config -D /var/lib/evebox users add --username jason \
    --github-username jasonish
```

You will be prompted with user details retrieved from GitHub and must provide confirmation before the user is actually added to the database.

GitHub must also be enabled in the configuration file with the `client-id`, `client-secret` and `callback` URLs configured.

Note: The `client-id` and `client-secret` are obtained from GitHub by registering a new application under your "Developer settings", currently https://github.com/settings/developers.

2.4 TLS

2.4.1 Starting the EveBox Server with TLS

Before TLS can be used a private key and certificate must be obtained. EveBox provides a tool to generate a self signed certificate if a certificate cannot be obtained through other means.

Enabling TLS on the Command Line

--tls
: Enables TLS.

--tls-cert FILE
: Specify the filename of the TLS certificate file.

--tls-key FILE
: Specify the filename of the TLS private key. May be ommitted if the certificate file is a bundle containing the key.

Example:

```
evebox --tls --tls-cert cert.pem --tls-key key.pem
```

Enabling TLS in the Configuration File

TLS can be enabled in the configuration file under `http.tls`:

```
http:
  tls:
    enabled: true
    certificate: /path/to/cert.pem
    key: /path/to/key.pem
```

2.4.2 Creating a Self Signed Certificate

EveBox ships with a tool to generate self signed TLS certificates.

Example:

```
evebox gencert -o evebox.pem
```

Full usage of `evebox gencert`:

```
Usage of gencert:
      --duration int            Duration that certificate is valid for in days␣
↪(default 365)
      --hostname string         Hostname or IP address (one or more, comma separated)
      --org string              Organization name (default "EveBox User")
  -o, --outputFilename string   Output file (eg. evebox.pem)
```

2.4.3 Lets Encrypt

EveBox supports self managing TLS certificates from Lets Encrypt if the following conditions are met:

- The server can listen on port 443 (automatically set with the `--letsencrypt` command line option.
- EveBox is reachable publically with a DNS hostname, as required by the Acme protocol.

Due to the requirement of being publically reachable this is probably not useful for most users.

Example

Say your EveBox host is reachable at "demo.evebox.org", you would start EveBox like:

```
evebox server --letsencrypt demo.evebox.org
```

This will start the EveBox server on port 443 with TLS certificates automatically provisioned from Lets Encrypt.

As this requires listening on port 443, you will need to make sure the user running EveBox has the ability to bind to port 443.

Note: On Linux a program may be given the ability to bind to a privileged port by setting the appropriate capability, for example:

```
setcap 'cap_net_bind_service=+ep' /usr/bin/evebox
```

CHAPTER 3

EveBox Agent

The EveBox "agent" is a tool for sending *eve* events directly to EveBox without the need for tools like *Filebeat* and/or *Logstash*. Events sent with the agent are handled by the EveBox server and stored in the database by the server.

3.1 Command Line Options

```
Usage of agent:
  -c, --config string   Configuration file
      --server string   EveBox server URL
      --stdout          Print events to stdout
  -v, --verbose         Be more verbose
```

3.2 Configuration File

CHAPTER 4

Elastic Search Importer (evebox esimport)

The EveBox "esimport" command can be used to import *eve* log files directly into Elastic Search. For most basic use cases it can be used as an alternative to *Filebeat* and/or *Logstash*.

EveBox "esimport" features:

- Continuous (tail -f style) reading of *eve* log files.
- Bookmarking of reads so reading can continue where it stopped during a restart.
- GeoIP lookups using the MaxMind GeoLite2 database if provided by the user.
- HTTP user agent parsing.
- One shot imports to send an *eve* log file to Elastic Search once.

4.1 Logstash Compatibility

EveBox *esimport* is fully compatible with Logstash and can be used in a mixed environment where some *eve* logs are being handled by *Logstash* and others by *esimport*. In this case you will want to use the **–index** option to set the index the same that *Logstash* is importing to.

4.2 Elastic Search Compatible

EveBox *esimport* can be used with Elastic Search version 2 and 5. If the configured *index* does not exist, *esimport* will create a *Logstash 2* style template for *Elastic Search v2.x* and a *Logstash 5* style template for *Elastic Search v5.x* to maintain compatibility with *eve* events imported with *Logstash*.

4.3 Example Usage

4.3.1 Oneshot Import of an *Eve* Log File

The following example will send a complete eve.json to Elastic Search and exit when done:

```
evebox esimport --elasticsearch http://10.16.1.10:9200 --index logstash \
    --oneshot -v /var/log/suricata/eve.json
```

4.3.2 Continuous Import

This example will run *esimport* in continuous mode sending events to Elastic Search as they appear in the log file. The last read location will also be bookmarked so *esimport* can continue where it left off after a restart. For many use cases this can be used instead of *Filebeat* and/or *Logstash*.

```
./evebox esimport --elasticsearch http://10.16.1.10:9200 --index logstash \
    --bookmark --bookmark-filename /var/tmp/eve.json.bookmark \
    /var/log/suricata/eve.json -v
```

If using *esimport* in this way you may want to create a configuration named **esimport.yaml** like:

```
input: /var/log/suricata/eve.json
elasticsearch: http://10.16.1.10:9200
index: logstash
bookmark: true
bookmark-filename: /var/tmp/eve.json.bookmark
```

Then run *esimport* like:

```
./evebox esimport -c esimport.yaml -v
```

4.4 GeoIP

While EveBox *esimport* can do geoip lookups it does not include a geoip database. The only supported database is the MaxMind GeoLite2 database, see http://dev.maxmind.com/geoip/geoip2/geolite2/ for more information.

Note: Many Linux distributions that have a geoip database package use the old format of the database, not the current version supported by MaxMind.

While the **–geoip-database** option can be used to point *esimport* at the datbase, the following paths will be checked automatically, in order:

- /etc/evebox/GeoLite2-City.mmdb.gz
- /etc/evebox/GeoLite2-City.mmdb
- /usr/local/share/GeoIP/GeoLite2-City.mmdb
- /usr/share/GeoIP/GeoLite2-City.mmdb

Note: MaxMind provides their own program to update the databases. See http://dev.maxmind.com/geoip/geoipupdate/

4.4.1 GeoIP Quickstart

If you just want to get quickly started with GeoIP you can download the database to a path that *esimport* will automatically detect, for example:

```
mkdir -p /etc/evebox
cd /etc/evebox
curl -OL http://geolite.maxmind.com/download/geoip/database/GeoLite2-City.mmdb.gz
```

4.5 Command Line Options

--force-template
: Like Logstash's *overwrite-template*, this option will always load the EveBox template into Elasticsearch.

  While this option is off by default, it is recommended if only using EveBox to add events to Elasticsearch. It is off by default to better work with Elasticsearch instances where the template is already managed by Logstash of Filebeat.

4.6 Configuration File

The esimport command can use a YAML configuration file covering most of the command line arguments.

```
# The eve log file to read. Only one allowed.
input: /var/log/suricata/eve.json

# Elastic Search URL
elasticsearch: http://10.16.1.10:9200

# Elastic Search username and password.
#username: admin
#password: password

# Elastic Search index. -%{YYYY.MM.DD) will be appended, so this is just the
# prefix.
index: logstash

# For loading the EveBox template (Logstash compatible) into
# Elasticsearch. It is recommended to turn this option on if only
# using EveBox to add events to Elasticsearch. Leave disabled if
# already using Logstash or Filebeat on the same index.
#
# Default: false
#force-template: false

# Disable TLS certificate check.
#disable-certificate-check: true
```

(continues on next page)

(continued from previous page)

```
# When no bookmark is present start reading at the end of the file.
end: true

# Enable bookmarking so esimport can continue reading from where it
# left off after a restart.
bookmark: true

# Set a filename to keep the bookmark in case esimport cannot write to
# the log directory.
#bookmark-filename: /var/tmp/eve.json.bookmark

# Change the amount of events to batch per bulk request.
#batch-size: 1000

# Location of Suricata rule files to add to events.
#rules:
#  - /etc/suricata/rules/*.rules

geoip:
  # GeoIP is enabled by default if a database can be found.
  disabled: false

  # Path to the database, if not set some standard locations are
  # checked.
  #
  # The database used is the MaxMind GeoLite2 database. See:
  #    http://dev.maxmind.com/geoip/geoip2/geolite2/
  # Quick setup:
  #    cd /etc/evebox
  #    curl -OL http://geolite.maxmind.com/download/geoip/database/GeoLite2-City.mmdb.
↪gz
  #
  #database-filename: /etc/evebox/GeoLite2-City.mmdb.gz
  #database-filename: /etc/evebox/GeoLite2-City.mmdb
```

CHAPTER 5

API

EveBox exposes an API to the web based frontend that may be useful for other purposes. While the API is not stable yet, this is an attempt to document endpoints that are somewhat stable.

5.1 GET /api/1/alerts

The *alerts* endpoint returns alert groupings as seen in the EveBox *Inbox*, *Escalated* and *Alerts* views. An individual alert group is considered to be a grouping of a signature id, source address and destination address with a count of the number of times that event occurred and a time range containing the oldest occurrence of the alert, and newest occurrence. Additionally, the most recent occurence of the alert is returned.

In SQL terms the grouping is like `GROUP BY signature_id, GROUP BY src_ip, GROUP BY dest_ip`.

5.1.1 Query Parameters

`time_range` `(or timeRange)`
: Time range to limit matching alerts to. Only alerts between 'now' and time_range ago will be returned.

  Examples:

  - Last minute: `60s`
  - Last hour: `3600s`
  - Last 24 hours: `86400s`

  At this time only the 's' unit is support for seconds.

  This paramet is not allowed with `min_ts` or `max_ts`.

`min_ts`
: Specify the minimum timestamp for the range of the query. Alerts occurence on this or after will be included.

max_ts
Specify the maximum timestamp for the range of the query. Alerts occurring before or on this time will be included.

tags
A list of tags that events must, or must not have. Tags are commented separated, and if prefixed with "-", only alerts not having that tag will be returned.

The EveBox *inbox* is made of alerts that have not been archived, so use the value "-evebox.archived". The *escalated* view is made of alerts that have the "evebox.escalated" tagged and would be queries with a value of "evebox.escalated".

query_string (or queryString)
Query string alerts must match. The format of the query string varies depending on the datastore used.

5.1.2 Response Format

```
{
  "alerts": [
    {
      "count": 82,
      "event": {
      "_id": "98ae9349-136e-11e7-bba7-d8cb8a1db3b2",
      "_index": "logstash-2017.03.28",
      "_score": null,
      "_source": {
        "@timestamp": "2017-03-28T04:25:37.808Z",
        ...
      },
      "maxTs": "2017-03-27T22:25:37.808514-0600",
      "minTs": "2017-03-26T23:07:22.539277-0600",
      "escalatedCount": 0
    },
    {
      ...
    }
  ]
}
```

5.1.3 Examples

Query the "inbox" for alerts occurring in the last 24 hours:

```
curl -G http://localhost:5636/api/1/alerts \
    -d time_range=60s \
    -d tags=-archived
```

Query the "escalated" view:

```
curl -G http://localhost:5636/api/1/alerts \
    -d tags=evebox.escalated
```

Query the "Alerts" view for all alerts in the last 24 hours:

```
curl -G http://localhost:5636/api/1/alerts \
    -d time_range=84600s
```

Query alerts for all groups in the last 24 hours containing the string "GPL ICMP_INFO":

```
curl -G http://localhost:5636/api/1/alerts \
    -d time_range=84600s -d query_string="ICMP_INFO"
```

Query for alert groups with a destination IP of 10.16.1.10 in the last day:

```
curl -G http://localhost:5636/api/1/alerts \
    -d time_range=84600s -d query_string="dest_ip:10.16.1.10"
```

- genindex
- search

Index

Symbols

C

M

Q

T