**Suricata - Feature #3105**

**Add kafka output**

08/05/2019 04:00 AM - sandy sun

<table>
<thead>
<tr>
<th>Status:</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>sandy sun</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>TBD</td>
</tr>
<tr>
<td>Effort:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

Add kafka output.

Conf like this:

```
- alert-json-log:
  enabled: yes
  filetype: kafka
  kafka:
    brokers: >
      xxx-kafka-online003:9092,
      xxx-kafka-online004:9092,
      xxx-kafka-online005:9092,
      xxx-kafka-online006:9092,
      xxx-kafka-online007:9092
    topic: nsm_event
    partitions: 5
  http: yes
```

**History**

**#1 - 08/08/2019 10:14 PM - Andreas Herz**

- Assignee set to sandy sun
- Target version set to TBD

Can you add a bit more details how this would work and what advantages it might have?

**#2 - 08/09/2019 03:36 AM - sandy sun**

Andreas Herz wrote:

Can you add a bit more details how this would work and what advantages it might have?

Hi, Andreas Herz

start with eve kafka output:
- Install librdkafka, e.g.: yum install librdkafka-devel
- Configure with --enable-rdkafka option when you need output eve with kafka.
- Modify suricata.yaml in eve section or independent section like following:

```
filetype: kafka
filename: eve.json
#prefix: "@cee: " # prefix to prepend to each log entry
# the following are valid when type: syslog above
#identity: "suricata"
#facility: local5
#level: Info ## possible levels: Emergency, Alert, Critical, ## Error, Warning, Notice, Info, Debug
```

06/12/2020
brokers: >
A.B.C.D:9092,
E.F.G.H:9092
topic: event
partitions: 5

- alert-json-log:
  enabled: yes
  filetype: kafka
  kafka:
    brokers: >
      xxx-kafka-online003:9092,
      xxx-kafka-online004:9092,
      xxx-kafka-online005:9092,
      xxx-kafka-online006:9092,
      xxx-kafka-online007:9092
topic: nsm_event
partitions: 5
http: yes

Bebefits:
1. no need use logstash
2. increase event throughput,
e.g. When used in IDC exit case or east-west xxgbps environment cause huge amount of events.
3. kafka is convenient for data analysis.

#3 - 08/09/2019 02:38 PM - Danny Browning
What happens when kafka is unavailable at startup? Do we assume it is temporarily down or fail startup?
What happens when kafka is unavailable while running (rebalance, etc.)? If retrying, how long do we buffer and retry for?
How much of this [https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md](https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md) will be exposed?
What happens when the user provides a bad configuration (e.g. max message size) for alerts that are produced?
How do we expose to the user that the event rate within suricata is exceeding the publish rate to kafka?

#4 - 08/18/2019 08:34 AM - sandy sun
Danny Browning wrote:

What happens when kafka is unavailable at startup? Do we assume it is temporarily down or fail startup?
What happens when kafka is unavailable while running (rebalance, etc.)? If retrying, how long do we buffer and retry for?
How much of this [https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md](https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md) will be exposed?
What happens when the user provides a bad configuration (e.g. max message size) for alerts that are produced?
How do we expose to the user that the event rate within suricata is exceeding the publish rate to kafka?

All theses cases refered are possible, but in most cases, used deafult librdkafka conf is enough.
I believe current suricata output methods, also can 100% promise no err and exception.
1. when kafka startup init failed, will log err msg and exit.
when temporarily down, depends on libradkafka reconnect, you can see socket.max.fails
2. librdkafka(as a client) conf can modify before install, normally used default value.
Also can set conf when init kafka ctx, i set queue.buffering.max.messages 500000.
3. bad config leads to init failed or send failed. If send failed(librdkafka self has already retried), will log err msg.
4. If produce rate exceed consumer rate, librdkafka's queue buffer may filled full.
I'm not sure whether librdkafka has rate limit.