Empirical Bootstrapping of EVE-JSON
Schema Documentation

Sascha Steinbiss, Konstantin Klinger
DCSO Deutsche Cyber-Sicherheitsorganisation GmbH
EUREF-Campus 22
10829 Berlin
https://www.dcs.de
info@dcs.de
About DCSO

- Managed (IT-)Security Service Provider
- Based in Berlin, Germany
- Founded by (and mostly for) German DAX 30 companies and scientific institutions
- Focus on advanced attack detection, mitigation and attacker profiling
- Cyber Defense Services
  - Strategic/Technical Threat Intelligence
  - Information/Identity Leakage Monitoring
  - Network Security Monitoring ("TDH")
  - Incident Response
  - ...
- TDH is long-time Suricata user with community ties
document all eve record types and fields

Added by Victor Julien 11 months ago. Updated about 1 month ago.

Status: Assigned
Priority: Normal
Assignee: Sascha Steinbiss
Target version: TBD
Affected Versions: medium
Difficulty: Label:

Description

For each document type, document fields and their types. Add examples.

It's probably best to add specific tickets for each of the record types.

Related issues

Related to Support #2685: SuriCon 2018 brainstorm
Related to Documentation #2620: Documentation: tagged_packets / event_type packet
### Before EVE-JSON: Text formats

#### fast.log

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Log Level</th>
<th>Message</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Priority</th>
<th>Classification</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Log Level</th>
<th>Message</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Priority</th>
<th>Classification</th>
</tr>
</thead>
</table>

#### dns.log and friends

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Log Level</th>
<th>Message</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Priority</th>
<th>Classification</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Log Level</th>
<th>Message</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Priority</th>
<th>Classification</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Log Level</th>
<th>Message</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Priority</th>
<th>Classification</th>
</tr>
</thead>
</table>
$ u2spewfoo unified2.alert.1570453464

(Event)
sensor id: 0  event id: 41  event second: 1454591580  event microsecond: 156852
sig id: 2200074  gen id: 1  revision: 2  classification: 26
priority: 3  ip source: 172.16.16.181  ip destination: 172.16.16.164
src port: 80  dest port: 60432  protocol: 6  impact_flag: 0  blocked: 0

Packet
sensor id: 0  event id: 41  event second: 1454591580
packet second: 1454591580  packet microsecond: 156852
linktype: 1  packet_length: 2962

[ 0] F8 16 54 F8 91 AC 00 0C 29 D8 DD 2B 08 00 45 00 ..T.....)...)....E.
[ 16] 0B 84 9C FB 40 00 40 06 18 FF AC 10 10 B5 AC 10 ....@.@.........
[ 32] 10 A4 00 50 EC 10 69 3B B7 C3 40 B8 B4 8F 80 10 ....P..i;@......
[ 48] 00 EB 84 F0 00 00 01 01 08 0A 00 EC 2B 3A 76 36 .............+:v6
[ 64] 52 5C 48 54 54 50 2F 31 2E 31 20 32 30 20 4F R\HTTP/1.1 200 0
[ 80] 4B 0D 0A 44 61 74 65 3A 20 54 68 75 2C 20 30 33 4K..Date: Thu, 04
[112] 30 30 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 00 GMT..Server:

...
[Oisf-users] Suricata 2.0 Available!

Victor Julien victor at milisc.net
Tue Mar 25 10:41:03 UTC 2014

- Previous message (by thread): [Oisf-users] suricata-2.0rc3 'make' error with Nvidia K20/Tesla
- Next message (by thread): [Oisf-users] Suricata 2.0 Available!
- Messages sorted by: [date] [thread] [subject] [author]

The OISF development team is proud to announce Suricata 2.0. This release is a major improvement over the previous releases with regard to performance, scalability and accuracy. Also, a number of great features have been added.

The biggest new features of this release are the addition of "Eve", our all JSON output for events: alerts, HTTP, DNS, SSH, TLS and (extracted) files; much improved VLAN handling; a detectionless 'NSM' runmode; much improved CUDA performance.

The Eve log allows for easy 3rd party integration. It has been created with Logstash in mind specifically and we have a quick setup guide here
https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Lagstash_Kibana_and_Suricata_JSON_output

*Download*

Get the new release here:
http://www.openinfosecfoundation.org/download/suricata-2.0.tar.gz

*Notable new features, improvements and changes*

- Eve log, all JSON event output for alerts, HTTP, DNS, SSH, TLS and files. Written by Tom Decanio of nPulse Technologies
- NSM runmode, where detection engine is disabled. Development supported by nPulse Technologies
- Various scalability improvements, clean ups and fixes by Ken Steel of Tilera
- Add --set command line option to override any YAML option, by Jason Ish of Emulex
[Oisf-users] Suricata 2.0 Available!

Victor Julien <victor at iminisc.net>
Tue Mar 25 10:41:03 UTC 2014

- Previous message (by thread): [Oisf-users] suricata-2.0rc3 'make' error with Nvidia K20/Tesla
- Next message (by thread): [Oisf-users] Suricata 2.0 Available!
- Messages sorted by: [date] [thread] [subject] [author]

The OISF development team is proud to announce Suricata 2.0. This release is a major improvement over the previous releases with regard to performance, scalability and accuracy. Also, a number of great features have been added.

The biggest new features of this release are the addition of "Eve", our all JSON output for events: alerts, HTTP, DNS, SSH, TLS and (extracted) files; much improved VLAN handling; a detectionless 'NSM' runmode; much improved CUDA performance.

The Eve log allows for easy 3rd party integration. It has been created with Logstash in mind specifically and we have a quick setup:

https://redmine.openinfosafoundation.org/projects/suricata/wiki/Eve_log_Suricata_JSON_output

*Download*

Get the new release:

http://www.openinfosecfoundation.org/download/suricata-2.0.tar.gz

*Notable new features, improvements and changes*

- Eve log, all JSON event output for alerts, HTTP, DNS, SSH, TLS and files. Written by Tom Decanio of npulse Technologies
- NSM runmode, where detection engine is disabled. Development supported by npulse Technologies
- Various scalability improvements, clean ups and fixes by Ken Steel of Tilera
- Add -set CommandLine option to override any YAML option, by Jason Ish of Emulex

*Eve log, all JSON event output for alerts, HTTP, DNS, SSH, TLS and files. Written by Tom Decanio of npulse Technologies*
EVE-JSON Example

```
{
  "timestamp": "2017-03-17T02:56:27.562994+0000",
  "flow_id": 2197033886090252,
  "pcap_cnt": 3291,
  "event_type": "alert",
  "src_ip": "173.247.245.85",
  "src_port": 80,
  "dest_ip": "192.168.1.46",
  "dest_port": 51704,
  "proto": "006",
  "community_id": "1:c0Va4aaivKFgZp5apSfVoTjq1Kw=",
  "tx_id": 0,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 2022962,
    "rev": 3,
    "signature": "ET CURRENT_EVENTS Evil Redirector Leading to EK Jul 12 2016"
  },
  "http": {
    "hostname": "www.keionline.org",
    "url": "/",
    "http_user_agent": "Mozilla/4.0",
    "http_content_type": "text/html",
    "http_refer": "http:\/\/www.bing.com/",
    "http_method": "GET",
    "protocol": "HTTP/1.1",
    "status": 200,
    "length": 10406,
    "http_response_body": "<span ....",
    "http_response_body": "PHNWfY4gc3R5bGU...8fkw2PE6C1AAA="
  },
  "app_proto": "http",
  "flow": {
    "pkts_toserver": 7,
    "pkts_toclient": 12,
    "bytes_toserver": 1172,
    "bytes_toclient": 11273,
    "start": "2017-03-17T02:56:23.263180+0000"
  },
  "payload": "SFRUUC8xLjEgVDPzuMzY+9EMZ6gi",
  "payload_printable": "HTTP/1.1 200 OK\r\n\n .........T3.......g.\"
},
```

6/26 October 30, 2019 Sascha Steinbiss, Konstantin Klinger
EVE-JSON: Current State of Documentation

- Opportunistic EVE-JSON format documentation on ReadTheDocs
- Documentation is not exhaustive
  - **Semantics**: not clear which fields exist, what they mean and how they depend on each other
  - **Syntax**: downstream users need to parse the format correctly and unambiguously
- Safety checks vs. structure definition
- Adaption of downstream processing tools (e.g. https://github.com/rhaist/surevego) necessary for each JSON structure change
Annotation Approach

Static checking won’t do

- Libjansson dynamically modifies JSON nodes in memory
- Non-trivial prediction of result structure using static source analysis

Empirical approach

- Use Suricata itself to create “real world” JSON output
- Use EVE-JSON results to derive schema → union across versions provides full field set
- Use field set to bootstrap per-field documentation to be integrated with Suricata’s ReadTheDocs
- Bug discovery not goal of this approach, but welcome side effect
Workflow Schemafication

Schemafication

JSON Schema

EVE

EVE

EVE

Versioned JSON Schema

http.hostname: 2.0 → 5.0
smb.function: 4.1.0 → 5.0
src_ip: 2.0 → 5.0
anomaly.code: 5.0 → 5.0
community_id: 4.1.0 → 5.0

Log Gathering

Log Gathering

2.0
3.0
5.0

Suricata

Suricata

2.0 3.0 ... 5.0

Input Preparation

Input Preparation

PCAPs

Rulesets

Configs

Doc Generation

Doc Generation

OISF

Suricata Docs

Version Integration

Version Integration

9/26 October 30, 2019 Sascha Steinbiss, Konstantin Klinger
Input: PCAPs

Sample collection

- Public pcap sets
  - Wireshark Wiki, MIT, Zeek, Chris Sanders, ICS/SCADA library, etc.
  - Manually curated samples

- Conversion to PCAP

- Intention: high diversity, covering typical anomalies

- 832 files, 934 MB, 5199084 packets

- Missing traffic samples can be added via GitHub PR
Input: Rulesets & Configs

Rulesets
- ET Open for Suricata 2.0/4.0/5.0
- *-events.rules
- Protocol-specific and filestore alerts

```
alert http any any -> any any (msg:"FOO HTTP"; threshold: type both, track by_src, count 5, seconds 60; sid:101;)
```

Configs
- Enable all protocols and additional output (e.g. X-Forwarded-For)
- Assume Rust is enabled and usage of latest possible version (e.g. DNS 1/2, filestore 1/2)
Workflow

Schemafication

EVE
EVE
EVE

Versioned JSON Schema

Community ID

RST template

Curator

Log Gathering

2.0
3.0
5.0

Suricata Docs

Integration

OISF

Doc Generation

PCAPs
Rule sets
Configs

http.hostname: 2.0 → 5.0
smb.function: 4.1.0 → 5.0
src_ip: 2.0 → 5.0
anomaly.code: 5.0 → 5.0
community_id: 4.1.0 → 5.0

12/26
October 30, 2019
Sascha Steinbiss, Konstantin Klinger
Log Gathering

1. Build Docker images for all EVE-enabled Suricata versions (≥ 2.0)
   - Docker images for all versions available from Docker Hub
   - Build scripts can be found on GitHub in each version’s build directory

2. Run every collected PCAP against each version and collect EVE (automated in parallel)

```bash
for pcap in `find /pcaps -type f -name '*.pcap*'`; do
    echo "$pcap"
    suricata -r "$pcap" -c /configs/suricata.yaml -l /logs/ -k none
done

echo "--simulate-ips with /pcaps/http_drop.pcap"
suricata -r /pcaps/http_drop.pcap -c /configs/suricata.yaml -l /logs/ -k none
```

↑ --simulate-ips

1. https://hub.docker.com/u/satta
Log Gathering: Docker Setup

```
satta/suricata-5.0
```

```
/pcaps
```

```
/rules
```

```
/config
```

```
/logs
```

```
---
```

```
suricata.log
```

```
eve.json
```

```
suricata-5.0
```

```
build-suricata-5.0.sh
```

```
suricata.yaml
```

```
eve.json.schema
```

```
all.rules
```

```
206_example_a.pcap
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```
Workflow

Schemafication

JSON Schema → EVE → 2.0 → JSON Schema

Log Gathering

2.0 SURICATA → 3.0 SURICATA → 5.0 SURICATA

Input Preparation

PCAPs → Rulesets → Configs

Doc Generation

OISF

Suricata Docs

Version Integration

http.hostname: 2.0 → 5.0
smb.function: 4.1.0 → 5.0
src_ip: 2.0 → 5.0
anomaly.code: 5.0 → 5.0
community_id: 4.1.0 → 5.0

Community

RST template

Curator
Schemafication

JSON Schema

- Vocabulary that enables annotation and validation of JSON documents
- Structure, value types, presence requirement, ...
- Specified in JSON itself

GenSON (https://github.com/wolverdude/genson)

- “Powerful, user-friendly JSON Schema generator built in Python.”
- Runs evidence JSON against seed schema (e.g. `{}`) and extends schema with each document seen
- Used to bootstrap one JSON schema for each version
Schemafication: Seed Adjustments

```
{
    "event_type": "stats",
    "stats": {
        "threads": {
            "RX#01": {
                "decoder": {
                    "pkts": 1,
                    "pkts_delta": 1,
                    ...
                }
            }
        }
    }
}
```

```
{
    "event_type": "alert",
    "pcap_cnt": 3100,
    ...
    "in_iface": "eth0",
    ...
}
```

```
{
    "type": "object",
    "properties": {
        "stats": {
            "type": "object",
            "properties": {
                "threads": {
                    "patternProperties": {
                        "\.*": {}
                    }
                }
            }
        }
    }
}
```
Workflow

Schemafication

- JSON Schema
- EVE
- Community ID

Doc Generation

- OISF
- Suricata Docs

Version Integration

- http.hostname
- smb.function
- src_ip
- anomaly.code
- community_id

Log Gathering

- Suricata
- PCAPs
- Rulesets
- Configs

Input Preparation

- EVE
- Community ID

Preparation

- Community ID
- RST template
- Curator

Integration

- Version

18/26 October 30, 2019 Sascha Steinbiss, Konstantin Klinger
Version Integration

- Merge all generated JSON schemas to create unified schema, annotating each field with supported version range
- *delta fields collapsed into their regular counterparts
- Field list used as basis to create ReStructured Text page hierarchy
  - Object→subobject structure maps to sections/subsections
  - *Outer template*: updated with each re-run – name, value type, versions. Imports...
  - *Inner template*: untouched when existing – contains curated content in named paragraphs
- Output
  - Sphinx compatible RST tree
  - General JSON schema?
Suricata 3.0
Introduces (net)flow, email, fileinfo, stats(-delta), tcp
227 fields (+174)
Suricata 3.2
Introduces dnp3, extends stats, tls
382 fields (+155)
Suricata 4.0
Introduces tunnel, extends stats
432 fields (+50)
Suricata 4.1
Introduces dhcp, ikev2, krb6, DNSv2, metadata, nfs, rpc, smb, tftp
752 fields (+320)
Format Evolution

Suricata 4.1.3
Extends stats
955 fields (+203)
Suricata 5.0
Introduces anomaly, rdp, sip, snmp, extends stats
1054 fields (+99)
Results: Bugs & Interesting Findings

Reduction in stats fields:

- Suricata 3.1.2 → 3.1.3:

  - 3whs_right_seq_wrong_ack_evasion
  - 3whs_right_seq_wrong_ack_evasion_delta

- Suricata 3.1.3 → 3.1.4:

  - 3whs_ack_in_wrong_dir
  - 3whs_ack_in_wrong_dir_delta
  - 3whs_async_wrong_seq
  - 3whs_async_wrong_seq_delta
Results: Bugs & Interesting Findings

Bug #1707 (malformed json if message is too big) in Suricata 3.0.0:

- Overly large JSON output can not be parsed by consuming software (e.g. jq, GenSON)
- Fixed with Suricata 3.0.1
- Responsible fields: payload, payload_readable, packet
- Output:

```python
ValueError: Expecting , delimiter: line 1 column 65537 (char 65536)
parse error: Invalid literal at line 1, column 65546
```

- Workaround:

```bash
cat eve.json | awk 'length($0) < 65500' > eve.json.clean
```
Results: Bugs & Interesting Findings

Bug #3216 (MSN protocol detection/parser is not working):
- No EVE-JSON output for MSN protocol detection since version 2.0
- MSN detection removed in Suricata 5.0

The case of the missing “metadata” event_type:
- No output for “metadata” event_type in EVE-JSON (Suricata 4.1.0–5.0.0)

```bash
# Metadata event type. Triggered whenever a pktvar is saved
# and will include the pktvars, flowvars, flowbits and
# flowints.
- metadata
```
Results: Bugs & Interesting Findings

Segmentation faults (Suricata 2.0.0 - 2.0.11):

- **Input**

  ```
  alert smtp any any -> any any (msg:"FOO FILESTORE SMTP"; filestore; sid:124;)
  ```

- **Output**

  ```
  [33] 10/10/2019 -- 12:21:05 - (suricata.c:983) <Notice> (SCPrintVersion) -- This is
  ↩  Suricata version 2.0dev (rev bc70fc0f7)
  ↩  -- [ERRCODE: SC_ERR_CONFLICTING_RULE_KEYWORDS(141)] - rule contains conflicting
  ↩  keywords.
  ↩  free(): double free detected in tcache 2
  ```

Bug in the rule parsing engine?
Results: community_id documentation
13.1.4.7. community_id

**Type**  string
First supported Suricata release  4.1.0
Latest supported Suricata release  5.0
Date Generated  Oct 25, 2019

**Description**

This field contains the Community ID (https://github.com/corelight/community-id-spec) for the flow this event is based on as base64-encoded string.

**Related configuration items**

```bash
# enable/disable the community id feature.
community-id: true
# Seed value for the ID output. Valid values are 0-65535.
community-id-seed: 42
```

The above configuration snippet is valid for Suricata versions 4.1.0 - 5.0.0.

**Example**

```json
{
   "timestamp": "2014-04-26T16:16:58.654077+0000",
   "flow_id": "254068186806141",
   "pcap_cnt": 4,
   "event_type": "alert",
   "src_ip": "192.168.18.50",
   "src_port": 56981,
   "dest_ip": "74.125.239.97",
   "dest_port": 443,
   "proto": "6",
   "community_id": "1:mE7Ze75S0sdvA7zTSzrULJU3zem",
}```
Next steps

Limitations

- Only one build/runtime configuration per version
- Non-scalar value type formatting in documentation
- Inefficient log gathering (huge amount of data)
- Unclear PCAP coverage

Open tasks

- Integration into RTD?
- Call for missing PCAP, feature and fields submission
- Code polishing
- Discuss community requirements

Who wants to help fill the gaps?

→ https://github.com/satta/suricata-json-schema
Questions?
Talk to us!

@ssatta
@kk_onstantin

Slides contain CC-BY Graphics from the Noun project: “rule” by akash khandavilli, “Box” by Creative Stall, “File” by newstudiodesign10, “Blueprint” by Jemis mali, “config” by Ige Maulana, “editor” by monkik