BETTER Schema Documentation

Release 1.0.0

David Wharton

Nov 13, 2019

Contents:

Better Enhanced Teleological and Taxonomic Embedded Rules Schema

CHAPTER 1

Summary

This document defines a schema and standard for embedding metadata in intrusion detection system (IDS) rules. The discussed metadata is composed as key-value pairs and is primarily intended to communicate teleological and taxonomic information about the rule in which it resides.

1.1 Background

1.1.1 Problem

As network bandwidth increases, attack methodologies expand, malicious traffic patterns fluctuate, and IDS ruleset sizes grow, the ability to programmatically understand the taxonomic and teleological characteristics of each IDS rule becomes invaluable. The decades-old practices of maintaining a rigid classification.config file and segregating rules into distinct files is onerous, not scalable, and, in many deployments, inviable for accurate ruleset tuning. Simply enabling all available IDS rules is rarely wise, prudent, or feasible for those concerned about rule performance, false-positives, volume, and value. There needs to be an easy way to "slice and dice" large rulesets so that they can be customized for each particular deployment.

1.1.2 Solution

Embed metadata key-value pairs in each rule that can be programmatically consumed to enable powerful ruleset optimization.

1.2 Schema

Contents

1.2.1 Version

This document defines version 1.0, released October 2019.

1.2.2 Scope

While the remainder of this document focuses on the `metadata` keyword supported by the Suricata and Snort IDS engines, its applicability should not be considered restricted to just those technologies, but can apply to rules for other IDS engines if they support similar capabilities.

1.2.3 Overview

The Suricata metadata keyword and Snort metadata keyword allow for non-functional (in terms of detection) information to be included (embedded) within a rule. The contents of the `metadata` keyword can be structured as comma separated key-value pairs.

1.2.4 Details

This schema defines a key-value pair structure in the `metadata` keyword for Suricata and Snort rules. The key-value pairs within the value of the `metadata` keyword are defined as comma separated, with a space separating the key and value, and the key being the first word.

Regarding the `metadata` keyword values:

- Key names and values are case insensitive and should be interpreted as such.
- Key names and values should only contain printable ASCII characters.
- Key names and values should be separated by a single space (ASCII 0x20).
- Whitespace before or after key names and key values should be ignored.
- Key names should only contain alphanumeric characters (A-Z, a-z, 0-9) and underscore ('_'); and should not start with a number.
- Key values must not contain commas (','), semicolons (';'), or double quotes ('"'), but may include spaces (' '), dashes ('-'), etc.
- Key values must not begin with '<' (ASCII 0x3C) or '>' (ASCII 0x3E).
- The key name "sid" is reserved and should not be used unless the value of the key is the same as that of the `sid` keyword in the rule.
- Characters, character locations, character combinations, etc. that are not supported by the IDS engine as values to the `metadata` keyword are implicitly not allowed.

A rules file should designate what schema and version its containing rules support. This should be specified in the file before any rules are specified, using the format:

```
<comment_character(s)>better-schema<space><version>
```

Example:

```
#better-schema 1.0
```

1.2.5 Keys and Values

This document attempts to canonize specific key names and, where it make sense, define a finite set of values or particular value format. The key names defined here should not be considered to be comprehensive and in fact, the use of custom keys is encouraged as long as they conform to this standard and do not conflict with the nomenclature and purpose of the keys already defined here. Ruleset creators are encouraged to implement as many of these keys as are applicable, although none are required.

Note that many keys can have multiple entries (i.e. logically, multiple values). This one-to-many relationship is not only allowed, but necessary to fully take advantage of the flexibility of this schema.

Defined keys

Table 1: BETTER Defined Keys

Key	Example Values	Notes
protocols	dcerpc dhcp dns ftp http icmp imap irc ldap ntp pop rpc sip smb smtp snmp ssh tcp telnet tftp tls udp vnc	**Protocol(s) the rule is attempting to inspect.** There is no distinction of type, function, layer, etc. Since it is generally assumed in this context, Internet Protocol (IP) is typically not included unless it is specified in the rule (e.g. `alert ip ...`) It is recommended that the protocol "TLS" include SSL and there not be a bifurcation having SSL and TLS.
attack_target	http-server http-client ftp-server tls-server dns-server sip-client database-server client server	**Defines what type asset is protected by this rule.** Suggested values follow the format of `<protocol>-server` or `<protocol>-client`, with <protocol> not including layer 4 and below, and common deviations including values like `database-server`.
mitre_attack	T1100 T1068 T1018 T1046	**MITRE ATT&CK Framework ID** https://attack.mitre.org/
capec_id	118 210 255	**CAPEC ID number** related to this rule. Only the integer value is used for key value. https://capec.mitre.org/
cwe_id	22 506 119	**CWE ID number** related to this rule. Only the integer value is used for key value. https://cwe.mitre.org/
malware	malware post-infection pre-infection download-attempt	**If a rule detects on malware traffic**, it should have a `malware` key (it may also have a malware related `cwe_id` and/or `capec_id` key). This is not designed to label specific malware or malware families, but to identify the rule as malware related and communicate broad malware function. See *Appendix A - malware metadata key value details* for details on example values.
cve	2015-0235 2019-10149	**CVE number** related to this rule. Value does not include leading 'CVE-' and maintains the dash ('-') between the year and sequence number. https://cve.mitre.org/
cvss_v2_base	7.5	**CVSS version 2 base score** for the vulnerability related

Note: The values shown for the `priority`, `hostile`, and `infected` keys are the complete list for those keys.

1.3 Examples

These examples help illustrate the concepts discussed in this document. Also, the structures in the Suricata EVE JSON log snippets show how the metadata key-value pairs should be logically interpreted.

1.3.1 Example 1

This `metadata` keyword in a rule:

```
metadata:cwe_id 20,cvss_v3_base 7.3,hostile src_ip,created_at 2019-06-01,capec_id 248,
↪updated_at 2019-06-11,
filename exploit.rules,priority medium,rule_source acme-rule-factory,cvss_v2_base 8.1,
↪attack_target server,
attack_target smtp-server,cvss_v3_temporal 7.1,cve 2019-91325,cvss_v2_temporal 7.9,
↪mitre_attack t1190,
protocols smtp,protocols tcp;
```

Results in this in the Suricata EVE JSON log:

```
{
  "metadata": {
    "protocols": [
      "tcp",
      "smtp"
    ],
    "mitre_attack": [
      "t1190"
    ],
    "cvss_v2_temporal": [
      "7.9"
    ],
    "cve": [
      "2019-91325"
    ],
    "cvss_v3_temporal": [
      "7.1"
    ],
    "attack_target": [
      "smtp-server",
      "server"
    ],
    "cvss_v2_base": [
      "8.1"
    ],
    "rule_source": [
      "acme-rule-factory"
    ],
    "priority": [
      "medium"
    ],
```

(continues on next page)

(continued from previous page)


```
    "filename": [
      "exploit.rules"
    ],
    "updated_at": [
      "2019-06-11"
    ],
    "capec_id": [
      "248"
    ],
    "created_at": [
      "2019-06-01"
    ],
    "hostile": [
      "src_ip"
    ],
    "cvss_v3_base": [
      "7.3"
    ],
    "cwe_id": [
      "20"
    ]
  }
}
```


1.3.2 Example 2

This `metadata` keyword in a rule:

```
metadata:cwe_id 507,malware post-infection,hostile dest_ip,created_at 2016-03-21,
↪updated_at 2016-04-02,
filename acme.rules,priority high,infected src_ip,rule_source acme-rule-factory,
↪attack_target http-client,
attack_target client,mitre_attack t1094,protocols http,protocols tcp;
```

Results in this in the Suricata EVE JSON log:


```
{
  "metadata": {
    "protocols": [
      "tcp",
      "http"
    ],
    "mitre_attack": [
      "t1094"
    ],
    "attack_target": [
      "client",
      "http-client"
    ],
    "rule_source": [
      "acme-rule-factory"
    ],
    "infected": [
      "src_ip"
    ],
```


(continues on next page)

(continued from previous page)

```
    "priority": [
      "high"
    ],
    "filename": [
      "acme.rules"
    ],
    "updated_at": [
      "2016-04-02"
    ],
    "created_at": [
      "2016-03-21"
    ],
    "hostile": [
      "dest_ip"
    ],
    "malware": [
      "post-infection"
    ],
    "cwe_id": [
      "507"
    ]
  }
}
```

1.4 Tools

1.4.1 Aristotle

- Aristotle is a Python script and library for the viewing and filtering of Suricata and Snort rulesets based on interpreted key-value pairs present in the metadata keyword within each rule.
- https://github.com/secureworks/aristotle

1.5 Authors

- David Wharton, Secureworks Counter Threat Unit

1.6 Appendices

1.6.1 Appendix A - `malware` metadata key value details

Value	Description
malware	Malware related traffic (generic)
post-infection	Malware post-infection
pre-infection	Malware pre-infection
download-attempt	Malware download attempt; pre-persistence

1.6.2 Appendix B - `priority` metadata key value details

Value	Details
high	High priority issues; typically reserved for malware infection and post-compromise traffic.
medium	Pre-infection; exploit attempts to download malware; targeted exploitation attempts
low	lower priority threats; scanning, etc.
info	Informational. Alert is generated/logged but is not significant enough on its own to warrant action.
research	Rule deployed for research purposes. Can and should be ignored by SIEM, analysts, etc.