Suricata Rule Taxonomy:

A Modest Teleological Approach

Secureworks®

COUNTER THREAT UNIT

David Wharton
Senior Security Researcher
SuriCon 2019

Problem

What Rules Should I Enable?

- Rules
  - Myriad rules available
- Sensor
  - Limited resources for sensor
    - VM, container, pod
    - 10, 20, 40, 100+ gigabit links
  - Particular assets being protected
  - Particular zone/environment
- Alerts
  - Only relevant alerts
  - Minimize false positives
  - Drop with confidence

Background

Previous Work (sort of)

- Intrusion Detection Message Exchange Format (IDMEF)
  - RFC 4765
  - Required "Classification" class
- Intrusion Detection Exchange Protocol (IDXP)
  - RFC 4767
  - Companion to IDMEF
- Incident Object Description Exchange Format (IODEF)
  - RFC 5070
  - Included "impact" class; "method" class
- Used/useful after alert has been generated (back end)
- Not useful on the front end (what rules to enable in the first place)

Background

Traditional Approach - `classtype` keyword in rule

- Requires maintaining corresponding `classification.config` file
- Broad, finite categories
  - Policy Violation (`policy-violation`)
  - Attack technique (`social-engineering`)
  - Attack target (`web-application-attack`)
  - Attack impact (`successful-dos`)
  - Traffic content classification (`inappropriate-content`)
  - Protocol usage (`icmp-event`, `non-standard-protocol`, `tcp-connection`)
  - Generic (`misc-activity`, `bad-unknown`, `unknown`, `not-suspicious`)

Background

Traditional Approach - `classtype` keyword in rule (continued)

- **Only one `classtype` rule option per rule!**
- **So which one to use?**
- **e.g. malware communication often falls into ALL of these categories:**
  - `tcp-connection`
  - `non-standard-protocol`
  - `malware-cnc`
  - `trojan-activity`
  - `unusual-client-port-connection`

Background

Traditional Approach - different `.rules` files

- Segregate rules into different `.rules` files … **same problems!**
  - Policy Violation (`policy.rules`)
  - Attack technique (`buffer-overflow.rules`)
  - Attack target (`web-attacks.rules`, `web_specific_apps.rules`)
  - Attack impact (`dos.rules`, `ddos.rules`)
  - Traffic content classification (`bad-traffic.rules`)
  - Protocols (`ftp.rules`, `smtp.rules`)
  - Generic (`misc.rules`, `bad-traffic.rules`, `other.rules`)
- **Can't have the same rules in multiple `.rules` files and have both files enabled!**
- e.g. should Exploit Kit detection go in `web_client.rules`, `exploit.rules`, `current_events.rules`, `browser-ie.rules`, `phishing.rules`, or `exploit-kit.rules`?

Need a BETTER approach...

BETTER Schema

Better Enhanced Teleological and Taxonomic Embedded Rules

- Key-Value pairs (with a teleological focus)
- Embedded into each rule
  - No external maintenance
- One-to-many relationships
- Implemented in `metadata` keyword
  - Bonus: Included in EVE (JSON) log!
- Can be programmatically consumed
- Flexible
  - Extend to fit your needs
- Simple

https://better-schema.readthedocs.io/

metadata Keyword

Simple Example

```
metadata: priority medium, protocols smtp,
protocols tcp;
```

EVE output

```
"metadata": {
  "protocols": [
    "tcp",
    "smtp"
  ],
  "priority": [
    "medium"
  ],
}
```

What key-value pairs to include?

What is useful?

- What are we trying to protect?
  - Data
    - But it has to pass thru, travel to, and live somewhere ... assets
  - Assets
    - Ultimate Endpoints
      - e.g. Server, Workstation, etc.
    - Subordinate Endpoints
      - e.g. Networking gear – Routers, Switches, etc.
    - Defined by what they communicate and how they communicate – *services* & *protocols*
- "Industry vertical"? … not useful … world is flat … industry attack surface is flat
  - A few exceptions, e.g. Operational Technology ("OT")
- What data do I already have?

BETTER Schema

Ground Rules

- Key names and values are case insensitive and should be interpreted as such.
- Key names and values should be 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 consist of 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.

Key names and values:

- **Case insensitive**
- **Printable ASCII**
- **Separated by a single space**
- **Key name is first word**
- **Key name can't include dash ('-')**

BETTER

Standard

- **Key Names**
  - `mitre_attack` vs `mitre_attck`
  - `protection_target` vs `attack_target`
  - `priority` vs `severity`
- **Key Values**
  - Finite list
    - `priority` → [“high”, “medium”, “low”, “info”, “research”]
  - Format
    - Dates, e.g. `created_at` → YYYY-MM-DD vs YYYY-DD-MM
    - `cve` → YYYY-NNNN vs CVE-YYYY-NNNN

BETTER Schema

Standard

Key	Example values	Notes
`protocols`	dhcp dns ftp http icmp imap ntp pop rpc sip smb smtp ssh tcp telnet tls udp	Protocol(s) the rule is attempting to inspect. There is no distinction of type, function, layer, etc. Since it is generally assumed in this application, Internet Protocol (IP) is not included unless it is specified in the rule (e.g. "alert ip ....") The protocol "tls" includes SSL; there should not be a bifurcation having SSL and TLS.

BETTER Schema

Standard

Key	Example values	Notes
`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. Typically in the format of "<protocol>-server" or "<protocol>-client", with <protocol> not including layer 4 and below. One notable addition is "database-server". "tls" includes SSL. Note that "tls-server" and "http-server" are distinct (same for "-client").

BETTER Schema

Standard

Key	Example values	Notes
`mitre_attack`	T1100 T1068	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/

BETTER Schema

Standard

Key	Example values	Notes
`malware`	malware post-infection pre-infection	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.
`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/

BETTER Schema

Standard

Key	Example values	Notes
`cvss_v2_base`	7.5 10.0	CVSS version 2 base score for the vulnerability related to this rule. https://www.first.org/cvss/v2/guide#2-1-Base-Metrics
`cvss_v2_temporal`	6.2 8.7	CVSS version 2 temporal score for the vulnerability related to this rule. https://www.first.org/cvss/v2/guide#2-2-Temporal-Metrics
`cvss_v3_base`	8.1 7.8	CVSS version 3.x base score for the vulnerability related to this rule. No differentiation of minor versions of CVSS v3 (e.g. 3.0 vs 3.1). https://www.first.org/cvss/v3.0/specification-document#2-Base-Metrics https://www.first.org/cvss/v3.1/specification-document#Base-Metrics
`cvss_v3_temporal`	7.7 7.9	CVSS version 3.x temporal score for the vuln related to this rule. No differentiation of minor versions of CVSS v3 (e.g. 3.0 vs 3.1). https://www.first.org/cvss/v3.0/specification-document#3-Temporal-Metrics https://www.first.org/cvss/v3.1/specification-document#Temporal-Metrics

BETTER Schema

Standard

Key	Example values	Notes
`priority`	high medium low info research	Corresponds directly with "priority" keyword in the Suricata rule: high = 1; medium = 2; low = 3; info = 4; research = 5.
`created_at`	2019-07-19 2017-10-31	Date the rule was created. Format is YYYY-MM-DD (ISO 8601).
`updated_at`	2019-04-02 2018-12-07	Date the rule was last updated. Format is YYYY-MM-DD (ISO 8601).

Values shown for `priority` are the complete list for that key.

BETTER Schema

Standard

Key	Example values	Notes
`hostile`	src_ip dest_ip	Which side of the alert is considered "hostile" (i.e. attacker, C2, etc.) This is the inverse of the "target" Suricata rule keyword (https://suricata.readthedocs.io/en/suricata-4.1.4/rules/meta.html#target).
`infected`	src_ip dest_ip	Which side of the alert is the malware-infected host. Should only be present on malware-related rules.

Values shown are the complete list for these keys.

BETTER Schema

Standard

Key	Example values	Notes
`filename`	sw.rules adware.rules	If the ruleset was split into files, this would be the corresponding filename. Defined to help provide legacy compatibility mapping.
`classtype`	trojan-activity shellcode-detect policy-violation	Same as what is/would be found in the classtype rule keyword. Defined to help provide legacy compatibility mapping. https://suricata.readthedocs.io/en/latest/rules/meta.html?highlight=classification%20keyword#classtype http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node31.html#SECTION00446000000000000000

BETTER Schema

Standard

Key	Example values	Notes
`rule_source`	secureworks emerging-threats	Vendor name or other identifier to label the source, author, and/or curator of the rule.

BETTER Schema

Standard

Key	Example values	Notes
`<your_own>`	<custom>	Keys and values that make sense for you and your environment, or for consumers of your ruleset.

metadata Keyword

Full Example

```
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;
```

EVE output


```
"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"
  ],
  "capec_id": [
    "248"
  ],
  ...
```

Compatible Rulesets

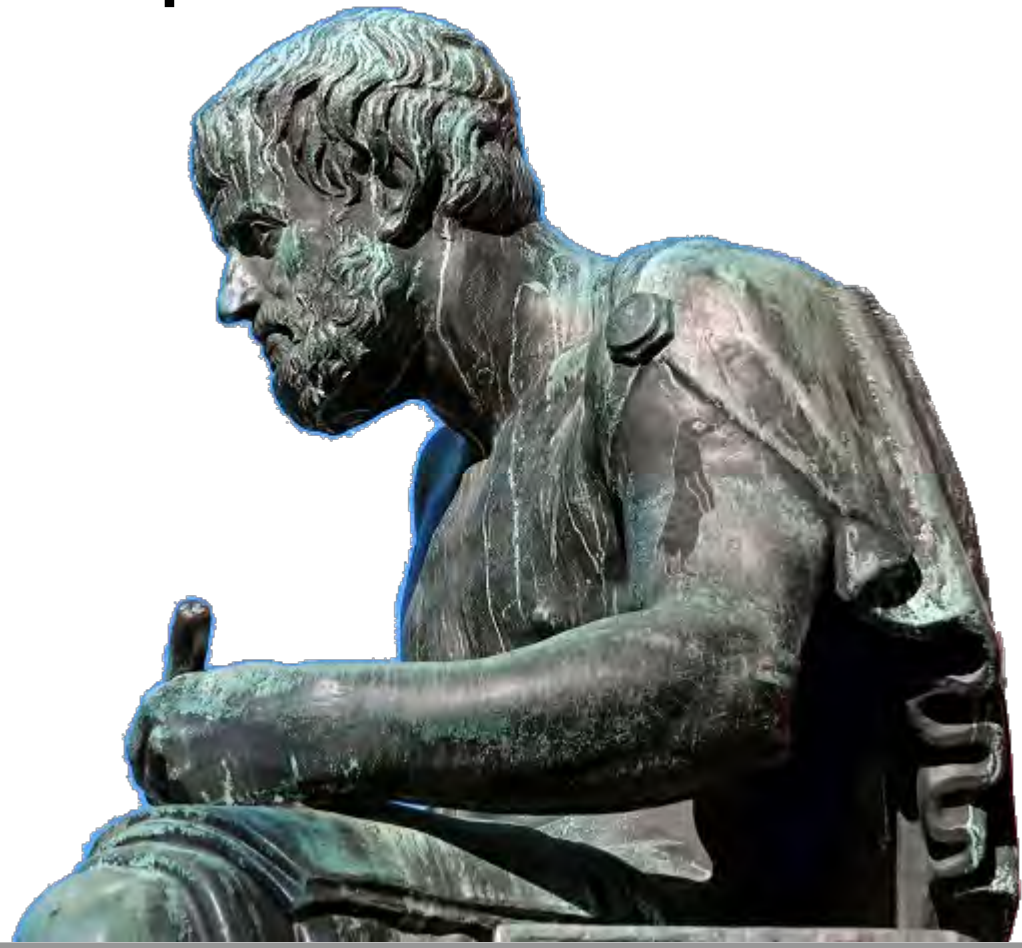
Where can I get BETTER rules?

- Encourage your ruleset vendor to add BETTER support!
  - New BETTER approach, adoption may take time
- Secureworks
  - “enhanced” Suricata ruleset
  - Fully compliant
  - Holistic coverage (all rules have all applicable key-value pairs)
- Emerging Threats
  - Mostly compliant
    - A few issues, e.g using underscores instead of dashes (e.g. dates)
  - Would benefit from better coverage
  - They are working on it (or at least considering it) – please encourage them!

Tools: Taking Advantage of BETTER Rulesets

Aristotle

- **Ruleset filtering based on metadata key-value pairs**
- Written in Python
  - Supports Python 2.7 and Python 3
- Stand alone script
- Library / Module
  - In PyPi
  - `pip install aristotle`
- Code
  - https://github.com/secureworks/aristotle/
- Docs
  - https://aristotle-py.readthedocs.io/



Tools: Aristotle

Powerful Boolean Filtering

- Boolean logic – uses the metadata key-value pairs as values in a (concrete) Boolean algebra
- Inputs
  - Ruleset (required)
  - Boolean “filter” string
- Outputs
  - Metadata statistics
  - Filtered Ruleset
    - Does not modify rules; only enable/disable
- Suricata-Update Support?
  - Yes, not in master branch yet … see `pull/209`

Aristotle

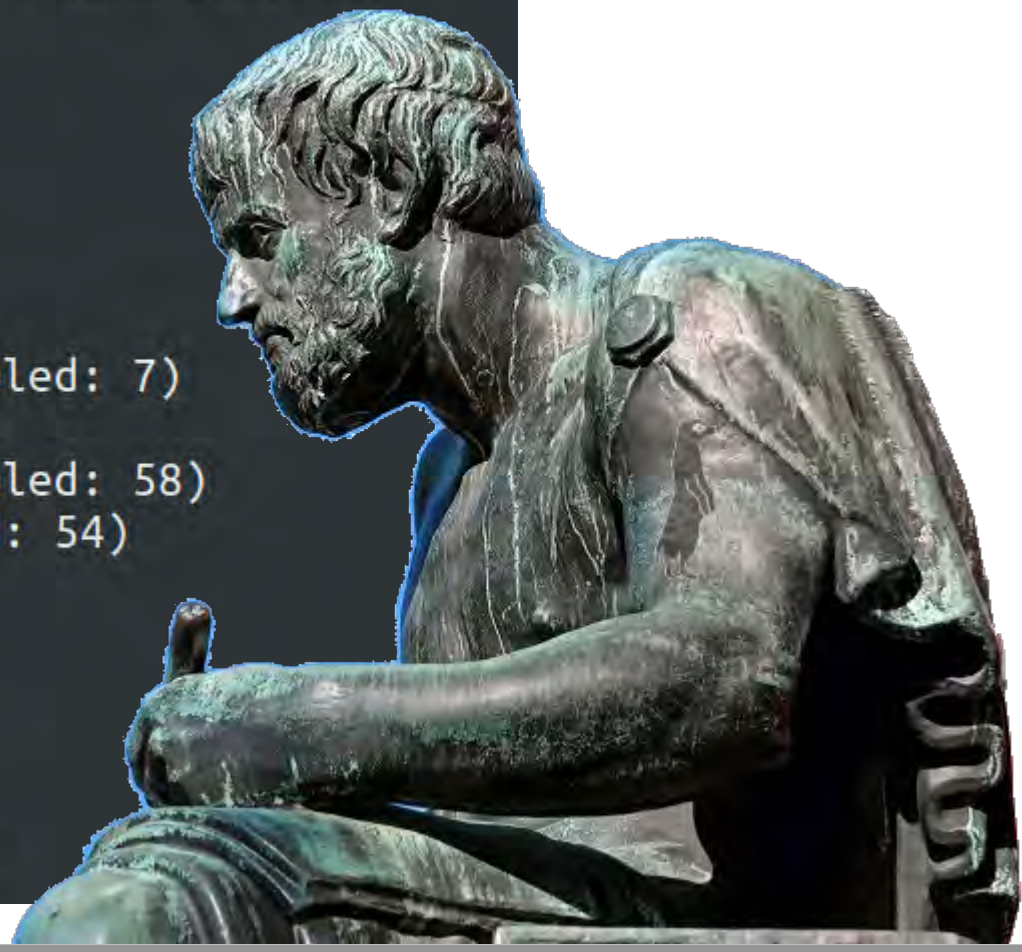
Statistics - keys

```
$ python aristotle.py --ruleset examples/example.rules --stats

        Aristotle
 Ruleset Metadata Tool

All Rules: Total: 6799; Enabled: 4977; Disabled: 1822

attack_target (Total: 6028; Enabled: 4554; Disabled: 1474)
malware (Total: 3467; Enabled: 3330; Disabled: 137)
cve (Total: 1570; Enabled: 887; Disabled: 683)
hostile (Total: 5962; Enabled: 4403; Disabled: 1559)
created_at (Total: 6799; Enabled: 4977; Disabled: 1822)
capec_id (Total: 2669; Enabled: 1191; Disabled: 1478)
updated_at (Total: 6799; Enabled: 4977; Disabled: 1822)
cwe_id (Total: 5199; Enabled: 4332; Disabled: 867)
priority (Total: 6799; Enabled: 4977; Disabled: 1822)
cvss_v3_base (Total: 271; Enabled: 259; Disabled: 12)
infected (Total: 2679; Enabled: 2520; Disabled: 159)
sid (Total: 6799; Enabled: 4977; Disabled: 1822)
cvss_v2_base (Total: 1130; Enabled: 829; Disabled: 301)
rule_source (Total: 6799; Enabled: 4977; Disabled: 1822)
cvss_v3_temporal (Total: 271; Enabled: 259; Disabled: 12)
filename (Total: 6799; Enabled: 4977; Disabled: 1822)
cvss_v2_temporal (Total: 1130; Enabled: 829; Disabled: 301)
protocols (Total: 6799; Enabled: 4977; Disabled: 1822)
```

Aristotle

Statistics – particular keys and values

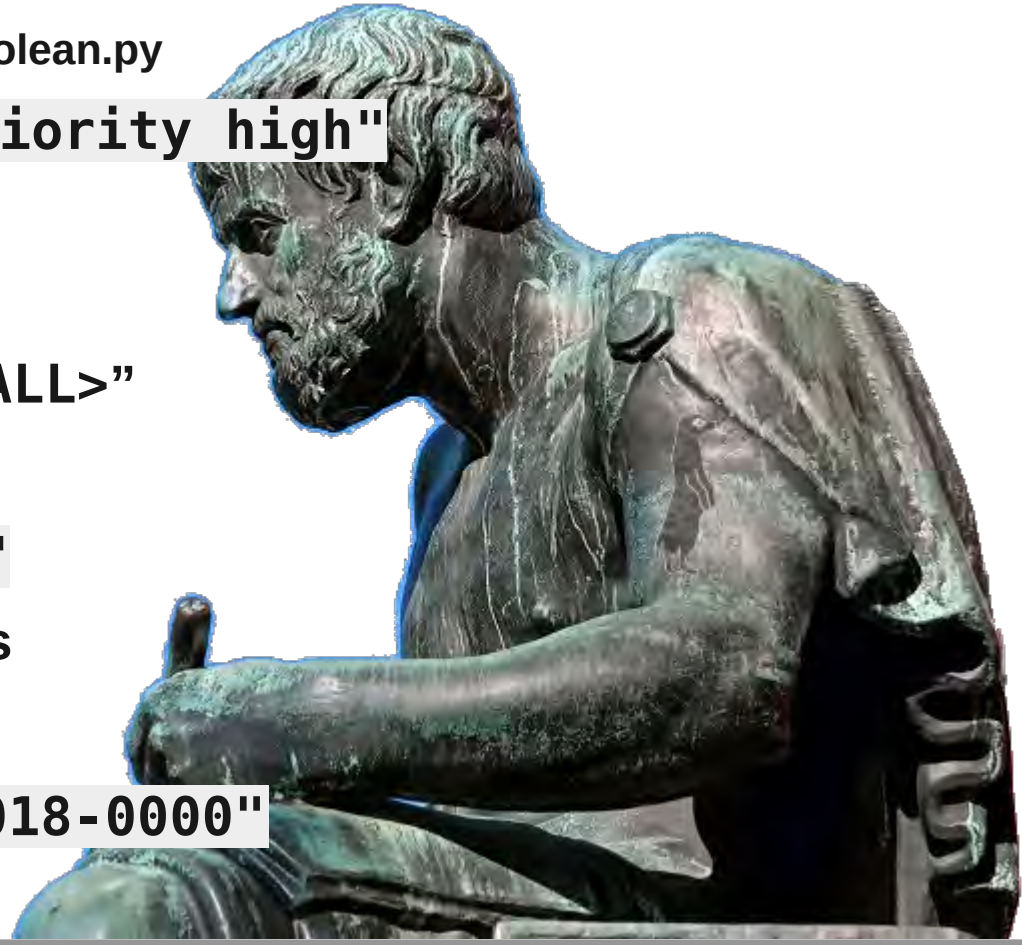
```
$ python aristotle.py -r examples/example.rules --stats malware priority

        Aristotle
 Ruleset Metadata Tool

All Rules: Total: 6799; Enabled: 4977; Disabled: 1822

malware (Total: 3467; Enabled: 3330; Disabled: 137)
        download-attempt (Total: 178; Enabled: 171; Disabled: 7)
        malware (Total: 135; Enabled: 117; Disabled: 18)
        post-infection (Total: 2647; Enabled: 2589; Disabled: 58)
        pre-infection (Total: 507; Enabled: 453; Disabled: 54)

priority (Total: 6799; Enabled: 4977; Disabled: 1822)
        high (Total: 2784; Enabled: 2752; Disabled: 32)
        info (Total: 375; Enabled: 170; Disabled: 205)
        medium (Total: 961; Enabled: 937; Disabled: 24)
        low (Total: 2679; Enabled: 1118; Disabled: 1561)
```



Aristotle

Boolean Filter String

- Boolean logic – AND, OR, NOT
  - Uses boolean.py module – https://github.com/bastikr/boolean.py
- Key-value pair surrounded by double quotes, e.g. `"priority high"`
- Group with parentheses ( )
- Extraneous whitespace (including newlines) ignored
- To match all values of a key, use the pseudo-value “<ALL>”
  - e.g. `"malware <ALL>"`.
- Can use the (pseudo) “sid” key, e.g. `"sid 80181444"`
- Applicable keys support the `>`, `<`, `>=`, and `<=` operators
  - `sid`, `cve`, `cvss_*`, `created_at`, `updated_at`
  - `"created_at >= 2019-01-01" OR "cve >= 2018-0000"`



Aristotle

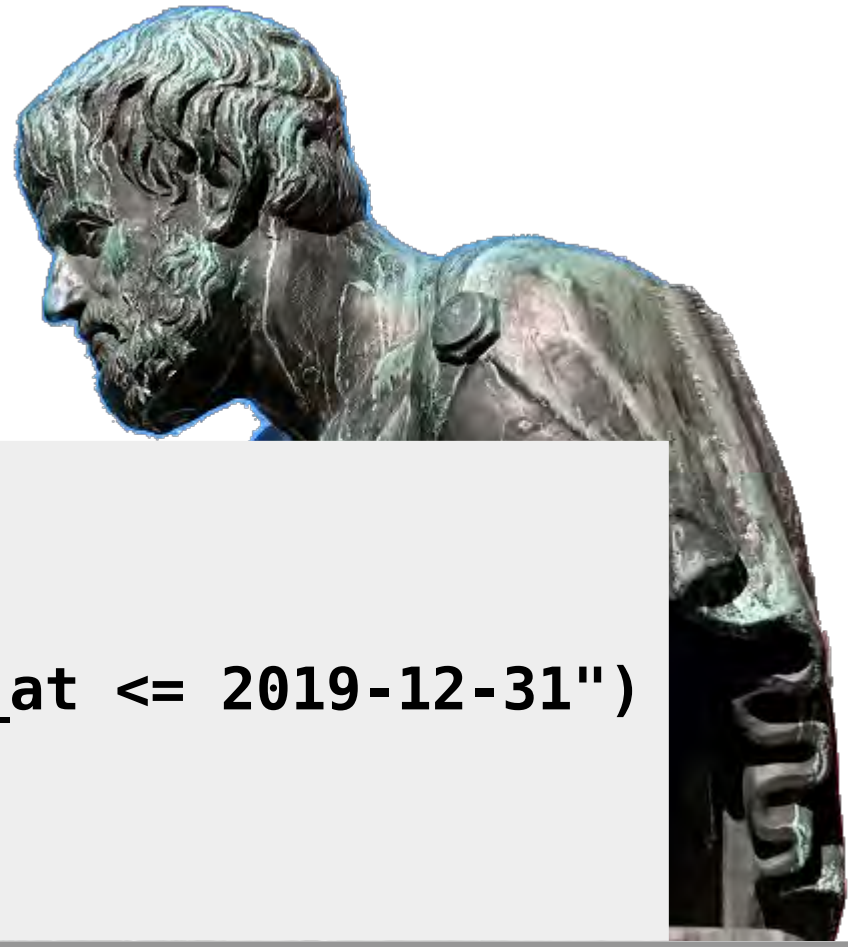
Boolean Filter String

- Match all high priority malware related rules:

```
"priority high" AND "malware <ALL>"
```

- Match all high priority rules that were created in 2019 or involve a vulnerability (based on CVE number) from 2018 or later:

```
 "priority high"
AND
 (
   ("created_at >= 2019-01-01" AND "created_at <= 2019-12-31")
  OR
   "cve >= 2018-0000"
 )
```



Aristotle

Filter: match all high and medium rules that are designed to protect a web server:

```
$ python3 aristotle.py -r examples/example.rules --filter '("priority high" OR "priority medium") AND
("attack_target http-server" OR "attack_target tls-server")' --summary --output suricata.rules

Acme - OLD MOTEL Exploitation Attempt Seen [sid:80186880]
Acme - SOUND ACCOMPANIST Exploitation Attempt Seen [sid:80186881]
Acme - VERY REPUNKNOWNMENT Exploitation Attempt Seen [sid:80185859]
Acme - CAUTIOUS DRAWING Exploitation Attempt Seen [sid:80185860]
Acme - OUTSTANDING SHOPPER Exploitation Attempt Seen [sid:80185861]
Acme - MATURE UNKNOWN Exploitation Attempt Seen [sid:80186885]
Acme - COLOSSAL PRIZEFIGHT Exploitation Attempt Seen [sid:80183304]
Acme - TINY STANDARD Exploitation Attempt Seen [sid:80183305]
Acme - ENVIOUS CLIP Exploitation Attempt Seen [sid:80185869]
Acme - ALTERUNKNOWN SODA Exploitation Attempt Seen [sid:80185870]
Acme - RELIEVED TIN Malware Communication [sid:80186382]
Acme - UNKNOWN SPIKE Malware Communication [sid:80186383]
Acme - FAMOUS COONSKIN Traffic Detected [sid:80186386]
Acme - SILENT TERRACOTTA Exploitation Attempt Seen [sid:80184858]
Acme - CRAZY INSURANCE Exploitation Attempt Seen [sid:80186396]
Acme - EFFECTIVE UNKNOWN Exploitation Attempt Seen [sid:80186397]

Showing 16 of 315 rules

Wrote 315 rules to file, 'suricata.rules'
```

Thank You

Secureworks

Distribution:

dwharton@secureworks.com

Questions

BETTER

`https://better-schema.readthedocs.io`

Aristotle

`https://github.com/secureworks/aristotle`