Aristotle Documentation

Release 1.0.2

David Wharton

Feb 24, 2021

Contents:

CHAPTER 1

Summary

Aristotle is a simple Python program that allows for the filtering of Suricata and Snort rulesets based on interpreted key-value pairs present in the metadata keyword within each rule. It can be run as a standalone script or utilized as a module.

1.1 Application Overview

Aristotle takes in a ruleset and can provide statistics on the included metadata keys. If a filter string is provided, it will also be applied against the ruleset and the filtered ruleset outputted.

Note: Aristotle does *not* modify the contents of rules. It simply includes or excludes rules based on the given Boolean filter string.

Aristotle is compatible with Python 2.7 and Python 3.x.

1.2 Background

Suricata and Snort support the `metadata` keyword that allows for non-functional (in terms of detection), arbitrary information to be included in a rule. By defining key-value pairs and including them in the metadata keyword, ruleset providers can embed rich teleological and taxonomic information. This information can be used to filter a ruleset – essentially enabling and disabling rules in a ruleset based on the metadata key-value pairs. Aristotle allows for the easy leveraging of the metadata key-value pairs to "slice-and-dice" Suricata and Snort rulesets that implement metadata key-value pairs.

1.3 Metadata Key-Value Pairs

Important: In order for Aristotle to be useful, it must be provided a ruleset that has rules with the metadata keyword populated with appropriate key-value pairs. Aristotle assumes that the provided ruleset conforms to the BETTER Schema.

1.4 Setup

Install dependencies:

```
pip install -r requirements.txt
```

Or if using as a module:

```
pip install aristotle
```

And refer to *Aristotle as a Module*

1.5 Usage

```
usage: aristotle.py [-h] -r RULES [-f METADATA_FILTER] [--summary]
                    [-o OUTFILE] [-s [STATS [STATS ...]]] [-i] [-q] [-d]

optional arguments:
  -h, --help            show this help message and exit
  -r RULES, --rules RULES, --ruleset RULES
                        path to rules file or string containing the ruleset
                        (default: None)
  -f METADATA_FILTER, --filter METADATA_FILTER
                        Boolean filter string or path to a file containing it
                        (default: None)
  --summary             output a summary of the filtered ruleset to stdout; if
                        an output file is given, the full, filtered ruleset
                        will still be written to it. (default: False)
  -o OUTFILE, --output OUTFILE
                        output file to write filtered ruleset to (default:
                        <stdout>)
  -s [STATS [STATS ...]], --stats [STATS [STATS ...]]
                        display ruleset statistics about specified key(s). If
                        no key(s) supplied, then summary statistics for all
                        keys will be displayed. (default: None)
  -i, --include-disabled
                        include (effectively enable) disabled rules when
                        applying the filter (default: False)
  -q, --quiet, --suppress_warnings
                        quiet; suppress warning logging (default: False)
  -d, --debug           turn on debug logging (default: False)
```

1.5.1 Example Files

The `examples` directory has `.filter` files that show examples of Boolean filter strings.

Also in the `examples` directory is an `example.rules` file that has a dummy Suricata ruleset that implements the BETTER Schema. While the example ruleset is syntactically correct, *it is not a real ruleset* intended to be used by a

Suricata sensor. It is provided to assist in demonstrating the functionality of Aristotle and to provide examples of rules with `metadata` keywords that conform to the BETTER Schema.

1.5.2 Example Usage

Note: `aristotle.py` in the root of the repository is a symlink to `aristotle/aristotle.py`. If the evironment in use does not recognize symlinks, adjust the paths accordingly.

Show high level statistics on all the keys in the `example.rules` file:

```
python aristotle.py -r examples/example.rules -s
```

Show statistics on the `protocols` key in the `example.rules` file:

```
python aristotle.py -r examples/example.rules -s protocols
```

Apply the Boolean filter defined in the `example1.filter` file against the rules in the `example.rules` file and output summary results to stdout:

```
python aristotle.py -r examples/example.rules -f examples/example1.filter --summary
```

Apply the Boolean filter defined in the `example1.filter` file against the rules in the `example.rules` file and output the results to the file `newrules.rules`:

```
python aristotle.py -r examples/example.rules -f examples/example1.filter -o newrules.
↪rules
```

Apply the Boolean filter defined specified on the command line against the rules in the `example.rules` file and output the results to the file `newrules.rules`:

```
python aristotle.py -r examples/example.rules -f '"malware <ALL>" AND ("attack_target␣
↪http-server" or "attack_target tls-server")' -o newrules.rules
```

Important: Because Aristotle requires key-value pairs (values) in the filter string to be enclosed in double quotes, a filter string specified on the command line must be enclosed in single quotes.

1.5.3 Statistics

The statistics command line option allows a user to to easily see what metadata key-value pairs the ruleset contains to assist in building a filter string.

If no key names are passed, summary info on all present keys is displayed:

```
$ python aristotle.py -r examples/example.rules -s

       Aristotle
 Ruleset Metadata Tool

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

  attack_target (Total: 6028; Enabled: 4554; Disabled: 1474)
```

(continues on next page)

(continued from previous page)

```
  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)
```

If one or more key names are passed, summary info is displayed for those keys:

```
$ python aristotle.py -r examples/example.rules -s malware protocols

       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)

protocols (Total: 6799; Enabled: 4977; Disabled: 1822)
    smtp (Total: 143; Enabled: 82; Disabled: 61)
    pop (Total: 64; Enabled: 45; Disabled: 19)
    rpc (Total: 16; Enabled: 4; Disabled: 12)
    dnp3 (Total: 5; Enabled: 0; Disabled: 5)
    vnc (Total: 1; Enabled: 0; Disabled: 1)
    ftp (Total: 130; Enabled: 65; Disabled: 65)
    sip (Total: 5; Enabled: 3; Disabled: 2)
    iccp (Total: 4; Enabled: 0; Disabled: 4)
    dns (Total: 20; Enabled: 6; Disabled: 14)
    ldap (Total: 1; Enabled: 1; Disabled: 0)
    irc (Total: 21; Enabled: 19; Disabled: 2)
    nntp (Total: 4; Enabled: 0; Disabled: 4)
    smb (Total: 60; Enabled: 42; Disabled: 18)
    http (Total: 5447; Enabled: 4199; Disabled: 1248)
    telnet (Total: 9; Enabled: 3; Disabled: 6)
    dcerpc (Total: 1; Enabled: 1; Disabled: 0)
    tcp (Total: 6788; Enabled: 4976; Disabled: 1812)
    imap (Total: 55; Enabled: 25; Disabled: 30)
    tls (Total: 145; Enabled: 128; Disabled: 17)
    modbus (Total: 7; Enabled: 0; Disabled: 7)
    tftp (Total: 1; Enabled: 0; Disabled: 1)
    ssh (Total: 9; Enabled: 4; Disabled: 5)
```

1.6 Boolean Filter Strings

A filter string defines the desired outcome based on Boolean logic, and uses the metadata key-value pairs as values in a (concrete) Boolean algebra:

- The Boolean operators `AND`, `OR`, and `NOT` are allowed.
- Grouping should be done with parentheses.
- **The key-value pair specifications must be surrounded by double quotes** (ASCII 0x22).
- **To match all values of a key**, use the pseudo-value "<ALL>" (not case sensitive), e.g. `"malware <ALL>"`.
- **To match a specific SID**, use the "sid" key, e.g. "sid 80181444", even though it may not be present in the `metadata` value.
  - A (pseudo) key of "sid" with the value of the rule's `sid` keyword is added to the internal key-value pair data structure(s).
  - If the ruleset `metadata` actually contains a "sid" key, it will be used instead of the value from the rule's `sid` keyword although if the values differ, a warning will be raised.
  - Note that per the BETTER Schema, a "sid" metadata key is not recommended but if present, it must have a value that matches the `sid` keyword value of the rule.
- Extraneous whitespace, including newlines, *is* allowed in the filter string.
- If a file containing a Boolean filter string is supplied:
  - Lines beginning with '#' are considered comments and are ignored.
  - A line starting with the string `<enable-all-rules>` results in enabling all rules, including disabled ones, before applying the Boolean filter.

The following keys support the >, <, >=, and <= operators in the filter string to specify, respectively, "greater than", "less than", "greater than or equal to", and "less than or equal to"; they must come between the key and value, and after the space that separates the key and value:

- `sid`
- `cve`
- `cvss_v2_base`
- `cvss_v2_temporal`
- `cvss_v3_base`
- `cvss_v3_temporal`
- `created_at`
- `updated_at`

1.6.1 Example Filter Strings

Match all high priority malware related rules:

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

Match all high priority malware related rules that were created in 2018 or later:

```
("priority high" AND "malware <ALL>") AND "created_at > 2018-01-01"
```

Match all high and medium rules that are designed to protect a webserver:

```
("priority high" OR "priority medium") AND ("attack_target http-server" OR
"attack_target tls-server")
```

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")
```

See more in the `examples` directory.

1.7 Aristotle as a Module

If the module is installed, Aristotle can be invoked from the command line and run like a script, e.g.:

```
python3 -m aristotle -r examples/example.rules --stats
```

Of course, Aristotle can be imported and used like a normal module:

```
import aristotle
```

For logging and/or output, attach to the logger named `aristotle` and add desired Handler(s), e.g.:

```
logger = logging.getLogger("aristotle")
logger.addHandler(logging.StreamHandler())
```

To use, create a `Ruleset` object and pass it a string containing the ruleset or a filename of a ruleset, along with a filter string. Then call the `Ruleset` object's `filter_ruleset()` function to get a list of SIDs matching the filter string.

Example:

```
import aristotle

a = aristotle.Ruleset("examples/example.rules")
a.set_metadata_filter("examples/example1.filter")
sids = a.filter_ruleset()
```

`Ruleset` class and functions:

class aristotle.**Ruleset**(*rules*, *metadata_filter=None*, *include_disabled_rules=False*, *summary_max=16*)

Class for ruleset data structures, filter string, and ruleset operations.

Parameters

- **rules** (*string, required*) – a string containing a ruleset or a filename of a ruleset file
- **metadata_filter** (*string, optional*) – A string or a filename of a file that defines the desired outcome based on Boolean logic, and uses the metadata key-value pairs as values in the Boolean algebra. Defaults to None (can be set later with `set_metadata_filter()`).
- **include_disabled_rules** (*boolean*) – effectively enable all commented out rules when dealing with the ruleset, defaults to *False*
- **summary_max** (*int, optional*) – the maximum number of rules to print when outputting summary/truncated filtered ruleset, defaults to *16*.

Raises *AristotleException*

filter_ruleset(*metadata_filter=None*)
Applies boolean filter against the ruleset and returns list of matching SIDs.

Parameters **metadata_filter** (*string, optional*) – A string that defines the desired outcome based on Boolean logic, and uses the metadata key-value pairs as values in the Boolean algebra. Defaults to `self.metadata_filter` which must be set if this parameter is not set.

Returns list of matching SIDs

Return type list

Raises *AristotleException*

get_all_sids()
Returns a list of all enabled SIDs.

Note: If `self.include_disabled_rules` is True, then all SIDs are returned.

Returns list of all enabled SIDs.

Return type list

get_stats(*key*, *keyonly=False*)
Returns string of statistics (total, enabled, disabled) for specified key and its values.

Parameters

- **key** (*string, required*) – key to print statistics for
- **keyonly** (*boolean, optional*) – only print stats for the key itself and not stats for all possible key-value pairs, defaults to *False*

Returns string contaning stats, suitable for printing to stdout

Return type string

Raises *AristotleException*

output_rules(*sid_list*, *outfile=None*)
Output rules, given a list of SIDs.

Parameters

- **sid_list** (*list, required*) – list of SIDs of the rules to output
- **outfile** (*string or None, optional*) – filename to output to; if None, output to stdout; defaults to *None*

Returns None

Return type NoneType

Raises *AristotleException*

print_header()
Prints vanity header and global stats.

print_ruleset_summary(*sids*)
Prints summary/truncated filtered ruleset to stdout.

Parameters **sids** (*list, required*) – list of SIDs.

Raises *AristotleException*

print_stats(*key*, *keyonly=False*)
Print statistics (total, enabled, disabled) for specified key and its values.

Parameters

- **key** (*string, required*) – key to print statistics for
- **keyonly** (*boolean, optional*) – only print stats for the key itself and not stats for all possible key-value pairs, defaults to *False*

set_metadata_filter(*metadata_filter*)
Sets the metadata filter to use.

Parameters **metadata_filter** (*string, required*) – A string or a filename of a file that defines the desired outcome based on Boolean logic, and uses the metadata key-value pairs as values in the Boolean algebra.

Raises *AristotleException*

1.8 License

Aristotle is licensed under the Apache License, Version 2.0.

1.9 Authors

- David Wharton

Index