

# Part 1









## # 1. Introduction and Objectives

### Welcome to this practical Suricata tutorial.

This guide walks through the process of installing, configuring, updating, testing, and customizing Suricata — a powerful open-source IDS/IPS and network security monitoring engine. The purpose of this project is to give hands-on experience with every critical step required to make Suricata operational in a real-world network environment.

### Objectives

By the end of this tutorial, you will be able to:

-  Navigate and understand the Suricata configuration directory structure
-  Load and manage Suricata's built-in and community-provided rule sets
-  Validate Suricata configurations using best practices
-  Start and monitor the Suricata service using systemd
-  Trigger and confirm alert generation with test payloads
-  Write, test, and refine custom Suricata rules
-  Analyze alerts using plaintext logs and structured `eve.json` logs
-  Troubleshoot rule syntax and engine startup issues effectively

## # 2 Directory Overview: /etc/suricata

This section introduces the contents of /etc/suricata, which is the main configuration directory for Suricata. Key files include suricata.yaml (the main configuration file), classification.config, reference.config, and threshold.config. A subdirectory called rules/ contains protocol-specific rule files. The command `ls -al /etc/suricata` is used to inspect the directory contents.

```
tiago-paquete@Linux:~$ ls -al /etc/suricata
```

```
=====
total 116
drwxr-xr-x  3 root root 4096 May  5 20:00 .
drwxr-xr-x 143 root root 12288 May  5 20:00 ..
-rw-r--r--  1 root root 3327 Feb  8 2024 classification.config
-rw-r--r--  1 root root 1375 Feb  8 2024 reference.config
drwxr-xr-x  2 root root 4096 May  5 20:00 rules
-rw-r--r--  1 root root 85175 Apr  1 2024 suricata.yaml
-rw-r--r--  1 root root 1643 Feb  8 2024 threshold.config
=====
```

## # 3. Rule File Structure: /etc/suricata/rules/

This chapter explores the /etc/suricata/rules/ directory, where Suricata stores various rule files that correspond to application-layer protocols and services (e.g., HTTP, DNS, TLS). The command `ls -al /etc/suricata/rules` lists all rule files. The chapter also includes a breakdown of a sample rule from `http-events.rules`, which demonstrates how HTTP anomalies are detected and alerts generated.

```
tiago-paquete@Linux:~$ ls -al /etc/suricata/rules
```

```
=====
total 152
drwxr-xr-x 2 root root 4096 May  5 20:00 .
drwxr-xr-x 3 root root 4096 May  5 20:00 ..
-rw-r--r-- 1 root root 1858 Feb  8 2024 app-layer-events.rules
-rw-r--r-- 1 root root 20880 Feb  8 2024 decoder-events.rules
-rw-r--r-- 1 root root  468 Feb  8 2024 dhcp-events.rules
-rw-r--r-- 1 root root 1221 Feb  8 2024 dnp3-events.rules
-rw-r--r-- 1 root root 1198 Feb  8 2024 dns-events.rules
-rw-r--r-- 1 root root 4005 Feb  8 2024 files.rules
-rw-r--r-- 1 root root  446 Feb  8 2024 ftp-events.rules
-rw-r--r-- 1 root root 14256 Feb  8 2024 http-events.rules
-rw-r--r-- 1 root root 3311 Feb  8 2024 http2-events.rules
-rw-r--r-- 1 root root 2832 Feb  8 2024 ipsec-events.rules
-rw-r--r-- 1 root root  585 Feb  8 2024 kerberos-events.rules
-rw-r--r-- 1 root root 2077 Feb  8 2024 modbus-events.rules
-rw-r--r-- 1 root root 2187 Feb  8 2024 mqtt-events.rules
-rw-r--r-- 1 root root  729 Feb  8 2024 nfs-events.rules
-rw-r--r-- 1 root root  558 Feb  8 2024 ntp-events.rules
-rw-r--r-- 1 root root  544 Feb  8 2024 quic-events.rules
-rw-r--r-- 1 root root  926 Feb  8 2024 rfb-events.rules
-rw-r--r-- 1 root root 4607 Feb  8 2024 smb-events.rules
-rw-r--r-- 1 root root 5393 Feb  8 2024 smtp-events.rules
-rw-r--r-- 1 root root  719 Feb  8 2024 ssh-events.rules
-rw-r--r-- 1 root root 14311 Feb  8 2024 stream-events.rules
-rw-r--r-- 1 root root 6861 Feb  8 2024 tls-events.rules
=====
```

```
tiago-paquete@Linux:~$ cat /etc/suricata/rules/http-events.rules
```

```
=====
# HTTP event rules
#
# SID's fall in the 2221000+ range. See http://doc.emergingthreats.net/bin/view/Main/SidAllocation
#
# These sigs fire at most once per HTTP transaction.
#
# A flowint http.anomaly.count is incremented for each match. By default it will be 0.
#
alert http any any -> any any (msg:"SURICATA HTTP unknown error"; flow:established; app-layer-
event:http.unknown_error; flowint:http.anomaly.count,+,1; classtype:protocol-command-decode;
sid:2221000; rev:1;)
...
=====
```

## # 4. Main Config: /etc/suricata/suricata.yaml

The Suricata YAML configuration file (/etc/suricata/suricata.yaml) is examined in this section. It explains how to define your network environment using variables such as HOME\_NET and EXTERNAL\_NET. It also covers interface settings (af-packet and pcap), the use of community-id, and the configuration for rule paths and logging. Commands like `sudo vim /etc/suricata/suricata.yaml` are used to inspect and edit the configuration.

```
tiago-paquete@Linux:~$ sudo vim /etc/suricata/suricata.yaml
```

```
=====
%YAML 1.1
```

```
---
```

```
# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata-yaml.html
```

```
# This configuration file generated by Suricata 7.0.3.
suricata-version: "7.0"
```

```
##
## Step 1: Inform Suricata about your network
##
```

```
vars:
```

```
# more specific is better for alert accuracy and performance
```

```
address-groups:
```

```
HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
```

```
#HOME_NET: "[192.168.0.0/16]"
```

```
#HOME_NET: "[10.0.0.0/8]"
```

```
#HOME_NET: "[172.16.0.0/12]"
```

```
#HOME_NET: "any"
```

```
EXTERNAL_NET: "!"$HOME_NET"
```

```
#EXTERNAL_NET: "any"
```

```
HTTP_SERVERS: "$HOME_NET"
```

```
SMTP_SERVERS: "$HOME_NET"
```

```
SQL_SERVERS: "$HOME_NET"
```

```
DNS_SERVERS: "$HOME_NET"
```

```
TELNET_SERVERS: "$HOME_NET"
```

```
AIM_SERVERS: "$EXTERNAL_NET"
```

```
DC_SERVERS: "$HOME_NET"
```

```
DNP3_SERVER: "$HOME_NET"
```

```
DNP3_CLIENT: "$HOME_NET"
```

```
MODBUS_CLIENT: "$HOME_NET"
```

```
MODBUS_SERVER: "$HOME_NET"
```

```
ENIP_CLIENT: "$HOME_NET"
```

```
ENIP_SERVER: "$HOME_NET"
```

```
...
```

```
vars:
```

```
# more specific is better for alert accuracy and performance
```

```
address-groups:
  HOME_NET: "[172.20.10.12/28]"
```

...

```
af-packet:
- interface: wlp0s20f3
  # Number of receive threads. "auto" uses the number of cores
  #threads: auto
  # Default clusterid. AF_PACKET will load balance packets based on flow.
```

...

```
pcap:
- interface: wlp0s20f3
  # On Linux, pcap will try to use mmap'ed capture and will use "buffer-size"
  # as total memory used by the ring. So set this to something bigger
  # than 1% of your bandwidth.
  #buffer-size: 16777216
```

...

```
# to make the id less predictable.
```

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

```
  # HTTP X-Forwarded-For support by adding an extra field or overwriting
```

...

```
##
```

```
default-rule-path: /var/lib/suricata/rules
```

```
rule-files:
- suricata.rules
```

```
##
```

...

```
=====
```

## # 5. Help and Version Verification

This short chapter covers how to verify the Suricata version and list its available command-line options. Since `--help` is not recognized, the usage output is viewed by simply running `suricata` without arguments. This output provides essential flags like `-c` for specifying the config file and `-T` for testing it.

```
tiago-paquete@Linux:~$ sudo suricata --help
```

```
=====
suricata: unrecognized option '--help'
```

```
Suricata 7.0.3
```

```
USAGE: suricata [OPTIONS] [BPF FILTER]
```

```

-c <path>                : path to configuration file
-T                        : test configuration file (use with -c)
-i <dev or ip>           : run in pcap live mode
```

```
...
=====
```

## # 6. Updating Rules with suricata-update

This chapter explains how to fetch and apply the latest detection rules from external sources using the `sudo suricata-update` command. By default, if no custom sources are set, the Emerging Threats Open ruleset is downloaded and installed into `/var/lib/suricata/rules/`. This step ensures Suricata can detect the latest known threats.

```
tiago-paquete@Linux:~$ sudo suricata-update
```

```
=====
6/5/2025 -- 09:40:52 - <Info> -- Using data-directory /var/lib/suricata.
6/5/2025 -- 09:40:52 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
6/5/2025 -- 09:40:52 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
6/5/2025 -- 09:40:52 - <Info> -- Found Suricata version 7.0.3 at /usr/bin/suricata.
6/5/2025 -- 09:40:52 - <Info> -- Loading /etc/suricata/suricata.yaml
6/5/2025 -- 09:40:52 - <Info> -- Disabling rules for protocol pgsq
6/5/2025 -- 09:40:52 - <Info> -- Disabling rules for protocol modbus
6/5/2025 -- 09:40:52 - <Info> -- Disabling rules for protocol dnp3
6/5/2025 -- 09:40:52 - <Info> -- Disabling rules for protocol enip
6/5/2025 -- 09:40:52 - <Info> -- No sources configured, will use Emerging Threats Open
6/5/2025 -- 09:40:52 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-7.0.3/
emerging.rules.tar.gz.
100% - 4904306/4904306
6/5/2025 -- 09:40:53 - <Info> -- Done.
```

...

```
=====
tiago-paquete@Linux:~$ sudo -al /var/lib/suricata/rules/
```

```
=====
total 36316
drwxr-xr-x 2 root root 4096 May 6 09:40 .
drwxr-xr-x 4 root root 4096 May 6 09:40 ..
-rw-r--r-- 1 root root 3228 May 6 09:40 classification.config
-rw-r--r-- 1 root root 37173983 May 6 09:40 suricata.rules
=====
```

## # 7. Managing Rule Sources

To expand rule coverage, Suricata allows updating, listing, and enabling new rule sources. The command `sudo suricata-update update-sources` fetches the index of available sources. You can list them with `sudo suricata-update list-sources` and enable a specific source (e.g. `aleksibovellan/nmap`) using `sudo suricata-update enable-source <name>`. After enabling, running `sudo suricata-update` again pulls rules from the added sources.

```
tiago-paquete@Linux:~$ sudo suricata-update update-sources
```

```
=====
6/5/2025 -- 09:45:45 - <Info> -- Using data-directory /var/lib/suricata.
6/5/2025 -- 09:45:45 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
6/5/2025 -- 09:45:45 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
6/5/2025 -- 09:45:45 - <Info> -- Found Suricata version 7.0.3 at /usr/bin/suricata.
6/5/2025 -- 09:45:45 - <Info> -- Downloading https://www.openinfosecfoundation.org/rules/
index.yaml
6/5/2025 -- 09:45:46 - <Info> -- Adding all sources
6/5/2025 -- 09:45:46 - <Info> -- Saved /var/lib/suricata/update/cache/index.yaml
=====
```

```
tiago-paquete@Linux:~$ sudo suricata-update list-sources
```

```
=====
6/5/2025 -- 09:46:13 - <Info> -- Using data-directory /var/lib/suricata.
6/5/2025 -- 09:46:13 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
6/5/2025 -- 09:46:13 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
6/5/2025 -- 09:46:13 - <Info> -- Found Suricata version 7.0.3 at /usr/bin/suricata.
Name: et/open
Vendor: Proofpoint
Summary: Emerging Threats Open Ruleset
License: MIT
Name: et/pro
Vendor: Proofpoint
Summary: Emerging Threats Pro Ruleset
License: Commercial
Replaces: et/open
Parameters: secret-code
Subscription: https://www.proofpoint.com/us/threat-insight/et-pro-ruleset
Name: oisf/trafficid
Vendor: OISF
Summary: Suricata Traffic ID ruleset
License: MIT
=====
```

```
tiago-paquete@Linux:~$ sudo suricata-update enable-source
aleksibovellan/nmap
```

```
=====
6/5/2025 -- 09:50:27 - <Info> -- Using data-directory /var/lib/suricata.
6/5/2025 -- 09:50:27 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
6/5/2025 -- 09:50:27 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
6/5/2025 -- 09:50:27 - <Info> -- Found Suricata version 7.0.3 at /usr/bin/suricata.
6/5/2025 -- 09:50:27 - <Info> -- Creating directory /var/lib/suricata/update/sources
6/5/2025 -- 09:50:27 - <Info> -- Enabling default source et/open
6/5/2025 -- 09:50:27 - <Info> -- Source aleksibovellan/nmap enabled
=====
```



tiago-paquete@Linux:~\$ sudo suricata-update

6/5/2025 -- 09:51:24 - <Info> -- Using data-directory /var/lib/suricata.

6/5/2025 -- 09:51:24 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml

...

## # 8. Testing the Suricata Configuration

Before running Suricata in live mode, it is essential to test the configuration. This is done using `sudo suricata -T -c /etc/suricata/suricata.yaml -v`. The output confirms if all rule files are correctly loaded and if there are any syntax errors or misconfigurations. A successful test shows how many rules were parsed and confirms that the YAML file is valid.

```
tiago-paquete@Linux:~$ sudo suricata -T -c /etc/suricata/suricata.yaml  
-v
```

```
=====
```

**Notice:** `suricata`: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode

**Info:** `cpu`: CPUs/cores online: 12

**Info:** `suricata`: Running suricata under test mode

**Info:** `suricata`: Setting engine mode to IDS mode by default

**Info:** `exception-policy`: master exception-policy set to: auto

**Info:** `logopenfile`: fast output device (regular) initialized: fast.log

**Info:** `logopenfile`: eve-log output device (regular) initialized: eve.json

**Info:** `logopenfile`: stats output device (regular) initialized: stats.log

**Info:** `detect`: 1 rule files processed. 43269 rules successfully loaded, 0 rules failed, 0

**Info:** `threshold-config`: Threshold config parsed: 0 rule(s) found

**Info:** `detect`: 43272 signatures processed. 1244 are IP-only rules, 4341 are inspecting packet payload, 37464 inspect application layer, 108 are decoder event only

**Notice:** `suricata`: Configuration provided was successfully loaded. Exiting.

```
=====
```

## # 9. Starting and Managing the Suricata Service

This section details how to control the Suricata daemon with systemctl. The service can be started with `sudo systemctl start suricata`, stopped with `sudo systemctl stop suricata`, and its status verified using `sudo systemctl status suricata`. If needed, a graceful shutdown can be performed with `sudo suricata -c shutdown`. Proper service management is key to deploying Suricata persistently on a system.

```
tiago-paquete@Linux:~$ sudo systemctl status suricata
```

```
=====
x suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: enabled)
   Active: failed (Result: exit-code) since Tue 2025-05-06 07:49:21 CEST; 2h 4min ago
 Duration: 154ms
    Docs: man:suricata(8)
          man:suricatasc(8)
          https://suricata.io/documentation/
 Main PID: 2204 (code=exited, status=1/FAILURE)
    CPU: 201ms
```

```
May 06 07:49:21 Linux systemd[1]: suricata.service: Scheduled restart job, restart counter is at 5.
```

```
May 06 07:49:21 Linux systemd[1]: suricata.service: Start request repeated too quickly.
```

```
May 06 07:49:21 Linux systemd[1]: suricata.service: Failed with result 'exit-code'.
```

```
May 06 07:49:21 Linux systemd[1]: Failed to start suricata.service - Suricata IDS/IDP daemon.
```

```
tiago-paquete@Linux:~$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
```

```
Notice: suricata: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode
```

```
Info: cpu: CPUs/cores online: 12
```

```
Info: suricata: Running suricata under test mode
```

```
Info: suricata: Setting engine mode to IDS mode by default
```

```
Info: exception-policy: master exception-policy set to: auto
```

```
Info: logopenfile: fast output device (regular) initialized: fast.log
```

```
Info: logopenfile: eve-log output device (regular) initialized: eve.json
```

```
Info: logopenfile: stats output device (regular) initialized: stats.log
```

```
Info: detect: 1 rule files processed. 43269 rules successfully loaded, 0 rules failed, 0
```

```
Info: threshold-config: Threshold config parsed: 0 rule(s) found
```

```
Info: detect: 43272 signatures processed. 1244 are IP-only rules, 4341 are inspecting packet payload, 37464 inspect application layer, 108 are decoder event only
```

```
Notice: suricata: Configuration provided was successfully loaded. Exiting.
```

```
tiago-paquete@Linux:~$ sudo systemctl start suricata.service
```

tiago-paquete@Linux:~\$ sudo systemctl status suricata.service

```
=====
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-05-06 09:57:13 CEST; 1min 24s ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata.io/documentation/
  Process: 9119 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile
/run/suricata.pid (code=exited, status=0/SUCCESS)
 Main PID: 9120 (Suricata-Main)
    Tasks: 18 (limit: 18529)
  Memory: 505.9M (peak: 506.3M)
    CPU: 24.613s
   CGroup: /system.slice/suricata.service
           └─9120 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/
suricata.pid
May 06 09:57:13 Linux systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
May 06 09:57:13 Linux suricata[9119]: i: suricata: This is Suricata version 7.0.3 RELEASE running
in SYSTEM mode
May 06 09:57:13 Linux systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.
=====
```

tiago-paquete@Linux:~\$ sudo systemctl stop suricata

tiago-paquete@Linux:~\$ sudo systemctl status suricata

```
=====
○ suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: ▶)
   Active: inactive (dead) since Tue 2025-05-06 12:05:53 CEST; 16s ago
 Duration: 2h 8min 38.085s
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata.io/documentation/
  Process: 9119 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
  Process: 11659 ExecStop=/usr/bin/suricatasc -c shutdown (code=exited, status=0/SUCCESS)
 Main PID: 9120 (code=exited, status=0/SUCCESS)
    CPU: 4min 10.553s

May 06 09:57:13 Linux systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
May 06 09:57:13 Linux suricata[9119]: i: suricata: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode
May 06 09:57:13 Linux systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.
May 06 12:05:51 Linux systemd[1]: Stopping suricata.service - Suricata IDS/IDP daemon.
May 06 12:05:51 Linux suricatasc[11659]: {"message": "Closing Suricata", "return_code": 0}
May 06 12:05:53 Linux systemd[1]: suricata.service: Deactivated successfully.
May 06 12:05:53 Linux systemd[1]: Stopped suricata.service - Suricata IDS/IDP daemon.
May 06 12:05:53 Linux systemd[1]: suricata.service: Consumed 4min 10.553s CPU time.
=====
```

### Optional graceful stop:

sudo suricatasc -c shutdown

## # 10. Logging Overview: /var/log/suricata/

Suricata writes various logs to /var/log/suricata, including eve.json (JSON-formatted alerts and metadata), fast.log (Snort-style alerts), stats.log (performance data), and suricata.log (general logs). The tail -f command is used to monitor logs in real-time, especially useful for viewing eve.json during tests.

```
tiago-paquete@Linux:~$ ls -al /var/log/suricata
```

```
=====
total 244
drwxr-xr-x  2 root root   4096 May  5 20:00 .
drwxrwxr-x 17 root syslog 4096 May  6 07:49 ..
-rw-r--r--  1 root root 150266 May  6 10:00 eve.json
-rw-r--r--  1 root root     0 May  5 20:00 fast.log
-rw-r--r--  1 root root  55146 May  6 10:00 stats.log
-rw-r--r--  1 root root  23349 May  6 09:57 suricata.log
=====
```

```
tiago-paquete@Linux:~$ tail -f /var/log/suricata/eve.json
```

## # 11. Simulating an Attack: NIDS Test

To validate detection, a simulated attack is triggered using `curl http://testmynids.org/uid/index.html`. This generates an alert, which can be confirmed in `fast.log`. This test demonstrates whether Suricata is actively inspecting traffic and generating alerts based on downloaded rules.

```
tiago-paquete@Linux:~$ curl http://testmynids.org/uid/index.html
```

```
=====
uid=0(root) gid=0(root) groups=0(root)
=====
```

```
tiago-paquete@Linux:~$ cat /var/log/suricata/fast.log
```

```
=====
05/06/2025-10:46:33.802418  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root
[**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP}
2600:9000:21c3:9400:0018:30b3:e400:93a1:80 ->
2a00:0020:0045:adab:23e9:cb3e:4f3a:07ea:39430
05/06/2025-10:46:39.494637  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root
[**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP}
2600:9000:21c3:4800:0018:30b3:e400:93a1:80 ->
2a00:0020:0045:adab:23e9:cb3e:4f3a:07ea:44108
=====
```

## # 12. Graceful Shutdown and Restart

This final section emphasizes safe control of the Suricata service. It shows how to stop and restart the IDS properly with `systemctl`, and provides an additional method for graceful shutdown using `suricatasc -c shutdown`. Graceful shutdown ensures all sessions and logs are closed correctly, avoiding data corruption or loss.

### Stop service:

```
tiago-paquete@Linux:~$ sudo systemctl stop suricata
```

### Restart:

```
tiago-paquete@Linux:~$ sudo systemctl restart suricata
```

### Optional debug stop:

```
tiago-paquete@Linux:~$ sudo suricatasc -c shutdown
```

# Part 2

## # Making and testing a custom rule:

```
tiago-paquete@Linux:~$ sudo vim /etc/suricata/rules/
```

```
=====
=====
=====
" Netrw Directory Listing                                     (netrw
v173)
"   /etc/suricata/rules
"   Sorted by      name
"   Sort sequence: [\/]$, \<core\%(\\.d\\+\\)\>=\>, \.h$, \.c$, \.cpp$,
\~\=\*$, *, \.o$, \.obj$, \.info$, \.swp$, \.bak$, \~$
"   Quick Help: <F1>:help  -:go up dir  D:delete  R:rename  s:sort-by
x:special
"
=====
=====
../
./
app-layer-events.rules
decoder-events.rules
dhcp-events.rules
dnp3-events.rules
dns-events.rules
files.rules
ftp-events.rules
http-events.rules
http2-events.rules
ipsec-events.rules
kerberos-events.rules
modbus-events.rules
mqtt-events.rules
nfs-events.rules
ntp-events.rules
quic-events.rules
rfb-events.rules
smb-events.rules
smtp-events.rules
ssh-events.rules
stream-events.rules
tls-events.rules
~
~
"/etc/suricata/rules/" is a directory
1,1          All
=====
```



## Making a custom rule:

```
tiago-paquete@Linux:~$ sudo vim /etc/suricata/rules/local.rules
=====
1 alert icmp any any -> $HOME_NET (msg: "ICMP Ping"; sid:1; rev:1;)
=====
```

## Rule Breakdown

Component	Meaning
alert	The <b>action</b> : trigger an alert when this rule matches.
icmp	The <b>protocol</b> to match: ICMP (used for ping, echo requests, etc.).
any any	<b>Source IP and port</b> : match any source IP and any port (port is ignored for ICMP).
->	Direction: from source to destination.
\$HOME_NET any	<b>Destination IP and port</b> : \$HOME_NET is a variable (typically your internal network), and any port is again ignored for ICMP.
(msg: "ICMP Ping"; sid:1; rev:1;)	Rule options:
- msg: "ICMP Ping"	The <b>alert message</b> to show in logs.
- sid:10000001	<b>Signature ID</b> — a unique identifier for the rule (should be >1000000 for custom rules to avoid conflicts).
- rev:1	<b>Revision number</b> — use this to track rule changes.

## Functional Summary

This rule triggers an alert when any ICMP packet is sent to your internal network (\$HOME\_NET) from any external source.

### In practice, this will mostly catch:

ICMP Echo Requests (ping)

Possibly other ICMP types (timestamp, address mask requests) depending on traffic

## Use Cases

**Detect ping scans** or basic network discovery tools (e.g. Nmap using ICMP).

**Alert on ICMP probes** from external sources to your internal network.

**Log and track ICMP traffic**, which is often overlooked in firewalls but may be used in reconnaissance.

## Considerations & Improvements

### Match only ICMP Echo Requests (Type 8):

```
alert icmp any any -> $HOME_NET any (msg: "ICMP Echo Request Detected"; icmp_type: 8; sid:1000001; rev:1;)
```

**Use a custom SID:** Always use sid >= 1000000 for custom rules to avoid conflicts with official rulesets.

### Rate limit or thresholding:

To avoid alert floods from continuous ping:

```
alert icmp any any -> $HOME_NET any (msg:"ICMP Echo Request"; icmp_type:8; sid:1000001; rev:2; threshold: type limit, track by_src, count 5, seconds 10;)
```

## Adding the rule path to the suricata.yaml file:

```
tiago-paquete@Linux:~$ sudo vim /etc/suricata/suricata.yaml
```

```
=====
```

```
...
```

```
##
```

```
## Configure Suricata to load Suricata-Update managed rules.
```

```
##
```

```
default-rule-path: /var/lib/suricata/rules
```

```
rule-files:
```

- suricata.rules
- /etc/suricata/rules/local.rules

```
##
```

```
## Auxiliary configuration files.
```

```
##
```

```
classification-file: /etc/suricata/classification.config
```

```
reference-config-file: /etc/suricata/reference.config
```

```
# threshold-file: /etc/suricata/threshold.config
```

```
##
```

```
## Include other configs
```

```
##
```

```
...
```

```
=====
```

## Testing:

```
tiago-paquete@Linux:~$ sudo suricata -T -c /etc/suricata/suricata.yaml
-v
=====
Notice: suricata: This is Suricata version 7.0.3 RELEASE running in
SYSTEM mode
Info: cpu: CPUs/cores online: 12
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Error: detect-parse: An invalid action "1" was given
Error: detect: error parsing signature "1 alert icmp any any ->
$HOME_NET (msg: "ICMP Ping"; sid:10000001; rev:1;)" from file /etc/
suricata/rules/local.rules at line 1
Info: detect: 2 rule files processed. 43269 rules successfully loaded, 1
rules failed, 0
Error: suricata: Loading signatures failed.
=====
```

## Problem

The number 1 at the beginning of the line is **not valid**. Suricata expects the rule to start directly with the **action**, such as alert, drop, reject, or pass.

## Solution

Edit the file /etc/suricata/rules/local.rules and **remove the leading 1**. The corrected rule should be:

```
alert icmp any any -> $HOME_NET any (msg: "ICMP Ping"; sid:10000001; rev:1;)
```

## Note:

- Add any for the destination port even for ICMP (for full syntax).
- Use sid:10000001 (custom rules should always use SIDs  $\geq$  1000000).

## After Correction: Test Again

Run the validation command again:

```
sudo suricata -T -c /etc/suricata/suricata.yaml -v
```

## Troubleshooting and retesting:

```
tiago-paquete@Linux:~$ sudo vim /etc/suricata/rules/local.rules
```

```
=====
alert icmp any any -> $HOME_NET any (msg: "ICMP Ping"; sid:10000001;
rev:1;)
~
=====
```

```
tiago-paquete@Linux:~$ sudo suricata -T -c /etc/suricata/suricata.yaml
-v
```

```
=====
Notice: suricata: This is Suricata version 7.0.3 RELEASE running in
SYSTEM mode
Info: cpu: CPUs/cores online: 12
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect: 2 rule files processed. 43270 rules successfully loaded, 0
rules failed, 0
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 43273 signatures processed. 1245 are IP-only rules, 4341
are inspecting packet payload, 37464 inspect application layer, 108 are
decoder event only
Notice: suricata: Configuration provided was successfully loaded.
Exiting.
=====
```

## Enable suricata.service and test with ping requests:

```
tiago-paquete@Linux:~$ sudo systemctl start suricata.service
```

```
tiago-paquete@Linux:~$ sudo systemctl status suricata
```

```
=====
● suricata.service – Suricata IDS/IDP daemon
   Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled;
  preset: enabled)
   Active: active (running) since Tue 2025-05-06 13:48:59 CEST; 11s
  ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata.io/documentation/
  Process: 12542 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/
suricata/suricata.yaml --pidfile /run/suricata.p
 Main PID: 12543 (Suricata-Main)
    Tasks: 1 (limit: 18529)
  Memory: 400.4M (peak: 400.4M)
     CPU: 12.003s
    CGroup: /system.slice/suricata.service
           └─12543 /usr/bin/suricata -D --af-packet -c /etc/suricata/
suricata.yaml --pidfile /run/suricata.pid

May 06 13:48:59 Linux systemd[1]: Starting suricata.service – Suricata
IDS/IDP daemon...
May 06 13:48:59 Linux suricata[12542]: i: suricata: This is Suricata
version 7.0.3 RELEASE running in SYSTEM mode
May 06 13:48:59 Linux systemd[1]: Started suricata.service – Suricata
IDS/IDP daemon.
=====
```

```
Tiagos-MacBook-Air ~ % ping 172.20.10.12
```

```
=====
PING 172.20.10.12 (172.20.10.12): 56 data bytes
64 bytes from 172.20.10.12: icmp_seq=0 ttl=64 time=70.319 ms
64 bytes from 172.20.10.12: icmp_seq=1 ttl=64 time=91.922 ms
...
64 bytes from 172.20.10.12: icmp_seq=36 ttl=64 time=58.902 ms
64 bytes from 172.20.10.12: icmp_seq=37 ttl=64 time=78.217 ms
^C
--- 172.20.10.12 ping statistics ---
38 packets transmitted, 38 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 12.836/63.813/119.858/29.849 ms
=====
```

```
tiago-paquete@Linux:~$ sudo systemctl stop suricata
```

tiago-paquete@Linux:~\$ sudo cat /var/log/suricata/fast.log

```
=====
05/06/2025-10:46:33.802418  [**] [1:2100498:7] GPL ATTACK_RESPONSE id
check returned root [**] [Classification: Potentially Bad Traffic]
[Priority: 2] {TCP} 2600:9000:21c3:9400:0018:30b3:e400:93a1:80 ->
2a00:0020:0045:adab:23e9:cb3e:4f3a:07ea:39430
05/06/2025-10:46:39.494637  [**] [1:2100498:7] GPL ATTACK_RESPONSE id
check returned root [**] [Classification: Potentially Bad Traffic]
[Priority: 2] {TCP} 2600:9000:21c3:4800:0018:30b3:e400:93a1:80 ->
2a00:0020:0045:adab:23e9:cb3e:4f3a:07ea:44108
05/06/2025-11:36:49.516620  [**] [1:2210044:2] SURICATA STREAM Packet
with invalid timestamp [**] [Classification: Generic Protocol Command
Decode] [Priority: 3] {TCP} 2606:4700:4400:0000:0000:0000:6812:202f:443
-> 2a00:0020:0045:adab:23e9:cb3e:4f3a:07ea:57378
05/06/2025-13:49:38.573886  [**] [1:10000001:1] ICMP Ping [**]
[Classification: (null)] [Priority: 3] {ICMP} 172.20.10.2:8 ->
172.20.10.12:0
05/06/2025-13:49:38.573968  [**] [1:10000001:1] ICMP Ping [**]
[Classification: (null)] [Priority: 3] {ICMP} 172.20.10.12:0 ->
172.20.10.2:0
=====
```

## Output Analysis

Relevant Alert Entries:

```
05/06/2025-13:49:38.573886  [**] [1:10000001:1] ICMP Ping [**] [Classification: (null)] [Priority: 3]
{ICMP} 172.20.10.2:8 -> 172.20.10.12:0
```

```
05/06/2025-13:49:38.573968  [**] [1:10000001:1] ICMP Ping [**] [Classification: (null)] [Priority: 3]
{ICMP} 172.20.10.12:0 -> 172.20.10.2:0
```

### Field Breakdown:

Field	Meaning
05/06/2025-13:49:38.573886	Timestamp of the packet.
[1:10000001:1]	Rule ID: Generator=1 (rule), SID=10000001 (your custom rule), Revision=1
ICMP Ping	Message you wrote in the rule (msg:).
Classification: (null)	No classification was assigned (optional config).
Priority: 3	Low priority alert (default when no classification is given).
{ICMP}	Protocol matched.
172.20.10.2:8 -> 172.20.10.12:0	ICMP Echo Request (Type 8) from MacBook to Suricata host.
172.20.10.12:0 -> 172.20.10.2:0	ICMP Echo Reply (Type 0) from Suricata host back to MacBook.

## Key Observations

### A. The Rule Is Working

Both incoming (Echo Request) and outgoing (Echo Reply) ICMP packets were matched by the rule.

The alert was triggered **twice per ping**: one for request, one for reply.

### B. Your Suricata Sensor Sees Bidirectional Traffic

This is important. It means:

Suricata is capturing on the correct interface.

Your interface is in promiscuous mode or properly routed.

### C. Classification Is Missing

You see Classification: (null) because the rule doesn't specify a classification.

To improve alerting logic and filtering, you can add one like this:

```
alert icmp any any -> $HOME_NET any (msg: "ICMP Ping"; sid:10000001; rev:1; classtype:icmp-
event;)
```

### D. Only Two Alerts Shown

You sent 38 pings, but only **2 alerts** are in the log.

Possible reasons:

You stopped Suricata before more packets were logged.

Alert suppression, rate limiting, or logging delays.

Suricata does not log duplicates if they occur too quickly unless you configure threshold or event\_filter.

## Recommendations

### Refine the rule for clarity and control:

```
alert icmp any any -> $HOME_NET any (msg: "ICMP Echo Request"; icmp_type:8; sid:10000001; rev:2; classtype:icmp-event;)
```

### Use threshold to prevent log flooding:

```
threshold: type limit, track by_src, count 1, seconds 5;
```

### Inspect with eve.json for structured data:

```
sudo jq '. | select(.alert)' /var/log/suricata/eve.json
```

### Keep Suricata running during tests, and always stop it gracefully if needed:

```
sudo systemctl restart suricata
```



# Setting a new improved custom rule and testing:

```
tiago-paquete@Linux:~$ sudo vim /etc/suricata/rules/local.rules
```

```
=====
alert icmp any any -> $HOME_NET any (msg: "ICMP Echo Request Detected";
icmp_type: 8; classtype: icmp-event; sid: 10000001; rev: 2; threshold:
type limit, track by_src, count 1, seconds 5;)
```

```
~
~
=====
```

## 1. Rule Header

This part defines the basic structure of the rule — what traffic it applies to and in what direction:

```
alert icmp any any -> $HOME_NET any
```

Part	Explanation
alert	<b>Action:</b> Generate an alert if the rule matches.
icmp	<b>Protocol:</b> Match ICMP traffic.
any	<b>Source IP:</b> Match any source address.
any	<b>Source port:</b> Required for syntax, even though ICMP doesn't use ports.
->	<b>Direction:</b> From source to destination.
\$HOME_NET	<b>Destination IP:</b> Typically your internal or protected network, defined in suricata.yaml.
any	<b>Destination port:</b> Also required by syntax, but unused for ICMP.

## Rule Options

Enclosed in parentheses (...), these options define what qualifies as a match and what metadata to attach to the alert.

### A. Match Condition

```
icmp_type: 8;
```

**Purpose:** This restricts matches to **ICMP Echo Requests**, i.e., what ping sends.

**Type 8:** Echo Request

**Type 0:** Echo Reply (not matched by this rule)

### B. Metadata and Alert Info

```
msg: "ICMP Echo Request Detected";
classtype: icmp-event;
sid: 10000001;
rev: 2;
```

Part	Explanation
msg:	<b>Message</b> that will appear in the alert logs. Useful for identifying what the alert means.
classtype:	Assigns a category from classification.config, helping organize and prioritize alerts.
sid:	<b>Signature ID:</b> Unique ID for the rule. Custom rules must use SID $\geq$ 1000000 to avoid conflicts with community/professional rules.
rev:	<b>Revision number:</b> Increments each time you update the rule (for version control and testing).

### C. Alert Suppression / Rate Limiting

threshold: type limit, track by\_src, count 1, seconds 5;

**Purpose:** Prevents log flooding from repeated pings.

Interpretation: For each source IP, trigger at most 1 alert every 5 seconds.

This avoids one alert per packet during high-frequency pings or ICMP scans.

Parameter	Meaning
type limit	Set a limit on how often the alert is triggered.
track by_src	Apply the threshold <b>per source IP</b> .
count 1	Allow <b>1 alert</b> per time window.
seconds 5	Time window duration (in seconds).

## Summary of Logical Flow

**Protocol Match:** ICMP packets.

**Type Match:** Echo Requests only (type 8).

**Address Match:** From any IP to any host inside \$HOME\_NET.

**Rate Limit:** No more than one alert per source IP per 5 seconds.

**Metadata:** Clearly labeled alert message, classification, and tracking info.

## Use Case Examples

**Detecting Reconnaissance:** ICMP Echo Requests are commonly used in network mapping or ping sweeps by attackers.

**Monitoring Activity:** Helps track which external hosts are actively probing your internal network.

**Training & Lab Use:** Safe and simple rule for validating Suricata setup and understanding custom rule creation.

## Test the Rule Load:

```
tiago-paquete@Linux:~$ sudo suricata -T -c /etc/suricata/suricata.yaml  
-v
```

```
=====
Notice: suricata: This is Suricata version 7.0.3 RELEASE running in
SYSTEM mode
Info: cpu: CPUs/cores online: 12
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Error: detect-parse: unknown rule keyword 'icmp_type'.
Error: detect: error parsing signature "alert icmp any any -> $HOME_NET
any (msg: "ICMP Echo Request Detected"; icmp_type: 8; classtype: icmp-
event; sid: 10000001; rev: 2; threshold: type limit, track by_src, count
1, seconds 5;)" from file /etc/suricata/rules/local.rules at line 1
Info: detect: 2 rule files processed. 43269 rules successfully loaded, 1
rules failed, 0
Error: suricata: Loading signatures failed.
=====
```

### The error message:

Error: detect-parse: unknown rule keyword 'icmp\_type'.

means that icmp\_type is not a valid keyword in Suricata rules — it's a Snort-specific option.

## Troubleshooting and retesting:

```
tiago-paquete@Linux:~$ sudo vim /etc/suricata/rules/local.rules
```

```
=====
alert icmp any any -> $HOME_NET any (msg: "ICMP Echo Request Detected";
itype: 8; classtype: icmp-event; sid: 10000001; rev: 2; threshold: type
limit, track by_src, count 1, seconds 5;)
=====
```

```
tiago-paquete@Linux:~$ sudo suricata -T -c /etc/suricata/suricata.yaml
-v
```

```
=====
Notice: suricata: This is Suricata version 7.0.3 RELEASE running in
SYSTEM mode
```

```
Info: cpu: CPUs/cores online: 12
```

```
Info: suricata: Running suricata under test mode
```

```
Info: suricata: Setting engine mode to IDS mode by default
```

```
Info: exception-policy: master exception-policy set to: auto
```

```
Info: logopenfile: fast output device (regular) initialized: fast.log
```

```
Info: logopenfile: eve-log output device (regular) initialized: eve.json
```

```
Info: logopenfile: stats output device (regular) initialized: stats.log
```

```
Info: detect: 2 rule files processed. 43270 rules successfully loaded, 0
rules failed, 0
```

```
Info: threshold-config: Threshold config parsed: 0 rule(s) found
```

```
Info: detect: 43273 signatures processed. 1244 are IP-only rules, 4341
are inspecting packet payload, 37464 inspect application layer, 108 are
decoder event only
```

```
Notice: suricata: Configuration provided was successfully loaded.
```

```
Exiting.
=====
```

```
tiago-paquete@Linux:~$ sudo systemctl restart suricata
```

```
tiagopaquete@Tiagos-MacBook-Air ~ % ping 172.20.10.12
```

```
=====
PING 172.20.10.12 (172.20.10.12): 56 data bytes
```

```
64 bytes from 172.20.10.12: icmp_seq=0 ttl=64 time=61.722 ms
```

```
64 bytes from 172.20.10.12: icmp_seq=1 ttl=64 time=78.364 ms
```

```
...
```

```
64 bytes from 172.20.10.12: icmp_seq=33 ttl=64 time=106.124 ms
```

```
64 bytes from 172.20.10.12: icmp_seq=34 ttl=64 time=23.192 ms
```

```
^C
```

```
--- 172.20.10.12 ping statistics ---
```

```
35 packets transmitted, 35 packets received, 0.0% packet loss
```

```
round-trip min/avg/max/stddev = 18.078/68.853/121.371/29.254 ms
=====
```

```
tiago-paquete@Linux:~$ sudo systemctl stop suricata
```

tiago-paquete@Linux:~\$ sudo cat /var/log/suricata/fast.log

```
=====
05/06/2025-10:46:33.802418  [**] [1:2100498:7] GPL ATTACK_RESPONSE id
check returned root [**] [Classification: Potentially Bad Traffic]
[Priority: 2] {TCP} 2600:9000:21c3:9400:0018:30b3:e400:93a1:80 ->
2a00:0020:0045:adab:23e9:cb3e:4f3a:07ea:39430
05/06/2025-10:46:39.494637  [**] [1:2100498:7] GPL ATTACK_RESPONSE id
check returned root [**] [Classification: Potentially Bad Traffic]
[Priority: 2] {TCP} 2600:9000:21c3:4800:0018:30b3:e400:93a1:80 ->
2a00:0020:0045:adab:23e9:cb3e:4f3a:07ea:44108
05/06/2025-11:36:49.516620  [**] [1:2210044:2] SURICATA STREAM Packet
with invalid timestamp [**] [Classification: Generic Protocol Command
Decode] [Priority: 3] {TCP} 2606:4700:4400:0000:0000:0000:6812:202f:443
-> 2a00:0020:0045:adab:23e9:cb3e:4f3a:07ea:57378
05/06/2025-13:49:38.573886  [**] [1:10000001:1] ICMP Ping [**]
[Classification: (null)] [Priority: 3] {ICMP} 172.20.10.2:8 ->
172.20.10.12:0
05/06/2025-13:49:38.573968  [**] [1:10000001:1] ICMP Ping [**]
[Classification: (null)] [Priority: 3] {ICMP} 172.20.10.12:0 ->
172.20.10.2:0
05/06/2025-14:52:03.220412  [**] [1:10000001:2] ICMP Echo Request
Detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP}
172.20.10.2:8 -> 172.20.10.12:0
05/06/2025-14:52:08.238054  [**] [1:10000001:2] ICMP Echo Request
Detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP}
172.20.10.2:8 -> 172.20.10.12:0
05/06/2025-14:52:13.255717  [**] [1:10000001:2] ICMP Echo Request
Detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP}
172.20.10.2:8 -> 172.20.10.12:0
05/06/2025-14:52:18.273176  [**] [1:10000001:2] ICMP Echo Request
Detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP}
172.20.10.2:8 -> 172.20.10.12:0
=====
```

## Output Analysis

Field	Value	Meaning
Timestamp	05/06/2025-14:52:03.220412	Date and time of the alert
[1:10000001:2]	Generator:SID:Revision = rule ID of your custom rule	
Message	"ICMP Echo Request Detected"	From msg: in your rule
Classification	"Generic ICMP event"	From classtype: icmp-event
Priority	3	Default priority for this classification
Protocol	{ICMP}	Detected protocol
Source IP and Type	172.20.10.2:8	Your MacBook sent ICMP Echo Request (type 8)
Destination IP and Type	172.20.10.12:0	Your Suricata-monitored host received it; 0 = placeholder for "port" (ICMP doesn't use ports)

## Why These Alerts Are a Success

The rule is now matching only ICMP **Echo Requests** (type 8).

The alerts are showing one alert every 5 seconds:

03, 08, 13, 18 — consistent with your threshold: type limit, track by\_src, count 1, seconds 5;

Suricata is correctly identifying and logging incoming ICMP ping packets.

The classification and priority fields are now present and readable, indicating your rule is fully integrated into Suricata's detection pipeline.

## Comparison with Previous Rule Behavior

Aspect	Old Rule	Updated Rule
Matched both Echo Requests and Replies	Yes	No (Echo Requests only)
Used classtype	No (null)	Yes (icmp-event)
Used thresholding	No	Yes (1 alert per 5 sec per source)
Used rev:1	Yes	Now rev:2 — ensures reloading works

# Part 3

## JSON Alerting in eve.json

```
tiago-paquete@Linux:~$ sudo apt-get install jq
```

```
=====
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.7.1-3build1).
jq set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 33 not upgraded.
=====
```

```
tiago-paquete@Linux:~$ sudo tail -f /var/log/suricata/eve.json | jq
'select(.event_type=="alert")'
```

```
=====
{
  "timestamp": "2025-05-06T14:52:13.255717+0200",
  "flow_id": 946663581833865,
  "in_iface": "wlp0s20f3",
  "event_type": "alert",
  "src_ip": "172.20.10.2",
  "src_port": 0,
  "dest_ip": "172.20.10.12",
  "dest_port": 0,
  "proto": "ICMP",
  "icmp_type": 8,
  "icmp_code": 0,
  "pkt_src": "wire/pcap",
  "community_id": "1:2A+P1vKKNWin1n9kny9Jh6/Nl7s=",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 10000001,
    "rev": 2,
    "signature": "ICMP Echo Request Detected",
    "category": "Generic ICMP event",
    "severity": 3
  },
  "direction": "to_server",
  "flow": {
    "pkts_toserver": 11,
    "pkts_toclient": 10,
    "bytes_toserver": 1078,
    "bytes_toclient": 980,
    "start": "2025-05-06T14:52:03.220412+0200",
    "src_ip": "172.20.10.2",

```

```
    "dest_ip": "172.20.10.12"
  }
}
{
  "timestamp": "2025-05-06T14:52:18.273176+0200",
  "flow_id": 946663581833865,
  "in_iface": "wlp0s20f3",
  "event_type": "alert",
  "src_ip": "172.20.10.2",
  "src_port": 0,
  "dest_ip": "172.20.10.12",
  "dest_port": 0,
  "proto": "ICMP",
  "icmp_type": 8,
  "icmp_code": 0,
  "pkt_src": "wire/pcap",
  "community_id": "1:2A+P1vKKNWin1n9kny9Jh6/Nl7s=",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 10000001,
    "rev": 2,
    "signature": "ICMP Echo Request Detected",
    "category": "Generic ICMP event",
    "severity": 3
  },
  "direction": "to_server",
  "flow": {
    "pkts_toserver": 16,
    "pkts_toclient": 15,
    "bytes_toserver": 1568,
    "bytes_toclient": 1470,
    "start": "2025-05-06T14:52:03.220412+0200",
    "src_ip": "172.20.10.2",
    "dest_ip": "172.20.10.12"
  }
}
```

=====



# 1. Detailed Explanation of eve.json (via jq)

## Top-Level Fields

Field	Description
timestamp	Exact time the alert was triggered (2025-05-06T14:52:13.255717+0200)
flow_id	Unique identifier for the flow Suricata is tracking
in_iface	Interface where packet was observed (wlp0s20f3, your Wi-Fi)
event_type	Type of event — in this case, alert
src_ip / dest_ip	IP addresses involved (MacBook → Suricata host)
src_port / dest_port	Always 0 for ICMP (no ports)
proto	Protocol matched (ICMP)
icmp_type	8 means Echo Request
icmp_code	0 — standard code for Echo Request

## alert Sub-Object

Field	Description
action	Suricata's action: allowed (didn't drop the packet)
gid	Generator ID (1 = Suricata rule)
signature_id	Your custom rule SID (10000001)
rev	Revision of the rule (2)
signature	The message from the rule: "ICMP Echo Request Detected"
category	From classtype: "Generic ICMP event"
severity	From classification config: 3 (low)

## flow Object

Field	Description
pkts_toserver / pkts_toclient	Number of packets in this flow, directionally split
bytes_toserver / bytes_toclient	Total bytes seen in each direction
start	When the flow started
src_ip / dest_ip	Redundant but useful for context

## 2. Comparison With fast.log

The equivalent fast.log entries:

05/06/2025-14:52:13.255717 **[\*\*]** [1:10000001:2] ICMP Echo Request Detected **[\*\*]** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 172.20.10.2:8 -> 172.20.10.12:0

Field	fast.log	eve.json
Time	05/06/2025-14:52:13.255717	" t i m e s t a m p " : "2025-05-06T14:52:13.255717+0200"
SID	[1:10000001:2]	"signature_id": 10000001, "rev": 2
Msg	ICMP Echo Request Detected	"signature": "ICMP Echo Request Detected"
Class	Generic ICMP event	"category": "Generic ICMP event"
Priority	Priority: 3	"severity": 3
Proto	{ICMP}	"proto": "ICMP"
Src/Dst	172.20.10.2:8 -> 172.20.10.12:0	"src_ip": "172.20.10.2", "icmp_type": 8

## 3. Which Format Should You Use and When

Use Case	Prefer
Quick human inspection	fast.log
Dashboard ingestion (e.g. ELK, Splunk, Wazuh)	eve.json
Alert correlation and post-processing	eve.json
Understanding flows, interfaces, or packet counts	eve.json
Lightweight logging in limited environments	fast.log