Содержание

Memory Issue Resolution 3
- ***mem_tracking_flow*** 3
- ***mem_preset*** 3
- ***mem_ssl_parsers*** 3
- ***mem_ssl_savebl*** 4
  - SSL parsing buffer save utilization statistics 4
- ***mem_quic_ietf_savebl*** 5

Memory Issue Resolution

Attention! Do not change these parameters on your own. Please contact technical support for assistance.

mem_tracking_flow

`mem_tracking_flow` — the amount of memory allocated for processing IPv4 flow when SSG starts.

Each processed flow will be placed in this memory and will be processed according to the policies and rules defined in the configuration. The occupancy of this pool can be monitored in `/var/log/dpi/fastdpi_stat.log` (as described here). If the pool is exhausted, SSG will no longer be able to process new flows.

mem_preset

`mem_preset` — the method of memory initialization in DPI.

DPI allocates memory statically: at the start of the process and when creating some service profiles (such as NAT, blacklists, and whitelists), additional memory is not allocated during operation. Memory consumption grows because the Linux OS distinguishes between resident (marked as RES in top) and virtual (marked as VIRT in top) memory of the process. The peculiarity is that until the memory is initialized (actually initialized with zero), it is not recorded by Linux as resident and is moved there as it is initialized.

Setting `mem_preset=1` in the `/etc/dpi/fastdpi.conf` file allows DPI to initialize almost all allocated memory at the start of the process. This prevents the growth of resident memory (RES) during operation, as the memory is fully initialized immediately. However, this mode slows down the system start and requires a sufficient amount of physical RAM. It is recommended to take this factor into account and monitor the consumption of virtual (VIRT) and resident (RES) memory.

mem_ssl_parsers

`mem_ssl_parsers` — the amount of memory allocated for SSL processing when SSG starts.

A clear sign that the current pool size is insufficient is the presence of errors in `/var/log/dpi/fastdpi_slave_*.log` such as:

```
[ERROR   ][000000118902699100][042E5F001EF5C480] Can't allocate record
ssl_state : IP : <IP:port> --> <IP:port>
[ERROR   ][000000118902954180][042E5F001EF5C50B] Can't allocate record
ssl_state_sni : IP : <IP:port> --> <IP:port>
```

A lack of SSL parsers can lead to problems in processing HTTPS traffic (which may affect filtering by RKN lists).

If such errors occur, it is recommended to increase the `mem_ssl_parsers` value by 1.5 - 2 times (considering the amount of free RAM).

mem_ssl_savebl

`mem_ssl_savebl` (cold) — specifies the number of buffers saved for SSL parsing when packets are shuffled.
Default = 50% of `mem_ssl_parsers`. If the value = `0`, saving and processing do not occur.
The first value is from the configuration file, and the value in parentheses is the one being used.

Example of output from alert:

1. Parameter not set

```
    mem_ssl_parsers                 : 320000
    mem_ssl_savebl                  : -1 (32000)
```

2. Parameter `mem_ssl_savebl=1234` is set

```
    mem_ssl_parsers                 : 320000
    mem_ssl_savebl                  : 1234 (1234)
```

SSL parsing buffer save utilization statistics

```
    [STAT    ][2024/08/19-17:26:05:599912] Detailed statistics on SSL_SAVEBL
:
             thread_slave= 0 : 1522/1/32000 0/0/0/0/0/ 6/6/2561 426/348/556
1/1/32000
             Total : 1522/1/32000 0/0/0/0/0/ 6/6/2561 426/348/556 1/1/32000
```

Legend: `a1/a2/a3 b1/b2/b3/b4/b5 c1/c2/c3 d1/d2/d3 e1/e2/e3`

a1 — size of allocated memory for saving records of subsequent parsing (matches snaplen)
a2 — records allocated
a3 — records used

b1 — total number of errors in processing saved packets
b2 — read buffer size is too large
b3 — invalid `isbl_t ind_` passed to the function
b4 — error adding records to arw — no space to save the list of used buffers
b5 — error adding data to `p_data` (unable to save the buffer)

c1 — number of data save requests
c2 — released saved packets
c3 — total size of packets that were saved

d1 — average size of saved TCP packets
d2 — minimum size of saved TCP packets
d3 — maximum size of saved TCP packets

e1 — records used in the arw queue
e2 — records available (can be reused)
e3 — records allocated in the queue

mem_quic_ietf_savebl

mem_quic_ietf_savebl — specifies the number of buffers for parsing quic_ietf requests, which consist of multiple packets. The default value is 15% of mem_ssl_parsers.