more detailed output about number of threads created

On a 8 core machine -

```bash
sudo suricata -S /dev/null -vvv --pcap=eth0 --set pcap.0.threads=auto --runmode=autofp --set threading.detect-thread-ratio=1
...
```

```bash
sudo suricata -S /dev/null -vvv --pcap=eth0 --set pcap.0.threads=auto --runmode=autofp --set threading.detect-thread-ratio=2
....
[9520] 11/10/2017 -- 20:45:46 - (flow-manager.c:828) <Config> (FlowManagerThreadSpawn) -- using 1 flow manager threads
[9520] 11/10/2017 -- 20:45:46 - (flow-manager.c:992) <Config> (FlowRecyclerThreadSpawn) -- using 1 flow recycler threads
[9520] 11/10/2017 -- 20:45:46 - (tm-threads.c:2178) <Notice> (TmThreadWaitOnThreadInit) -- all 17 packet processing threads, 4 management threads initialized, engine started.
```

I would expect the first run to have 8 packet processing threads and the second 16 packet processing threads.

Suricata info -

This is Suricata version 4.0.0-dev (rev 11806875)
Features: PCAP_SET_BUFF AF_PACKET HAVE PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAVE_HTP_URI_NORMALIZE_HOSTS PCRE_JIT HAVE_NSS HAVE LUA HAVE LUAJIT HAVE_LIBJANSSON TLS MAGIC
SIMD support: SSE_4_2 SSE_4_1 SSE_3
Atomic intrinsics: 1 2 4 8 16 byte(s)
64-bits, Little-endian architecture
GCC version 6.3.0 20170516, C version 199901
compiled with _FORTIFY_SOURCE=0
L1 cache line size (CLS)=64
thread local storage method: __thread
compiled with LibHTP v0.5.25, linked against LibHTP v0.5.25

Suricata Configuration:
- AF_PACKET support: yes
- PF_RING support: no
- NFQueue support: no
- NFLOG support: no

03/08/2022
<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPFW support</td>
<td>no</td>
</tr>
<tr>
<td>Netmap support</td>
<td>no</td>
</tr>
<tr>
<td>DAG enabled</td>
<td>no</td>
</tr>
<tr>
<td>Nspatech enabled</td>
<td>no</td>
</tr>
<tr>
<td>Unix socket enabled</td>
<td>yes</td>
</tr>
<tr>
<td>Detection enabled</td>
<td>yes</td>
</tr>
<tr>
<td>Libmagic support</td>
<td>yes</td>
</tr>
<tr>
<td>libnss support</td>
<td>yes</td>
</tr>
<tr>
<td>libnsspr support</td>
<td>yes</td>
</tr>
<tr>
<td>libjsonsson support</td>
<td>yes</td>
</tr>
<tr>
<td>hiredis support</td>
<td>no</td>
</tr>
<tr>
<td>hiredis async with libevent</td>
<td>no</td>
</tr>
<tr>
<td>Prelude support</td>
<td>no</td>
</tr>
<tr>
<td>PCRE jit</td>
<td>yes</td>
</tr>
<tr>
<td>LUA support</td>
<td>yes, through luajit</td>
</tr>
<tr>
<td>libluajit</td>
<td>yes</td>
</tr>
<tr>
<td>libgeoip</td>
<td>yes</td>
</tr>
<tr>
<td>Non-bundled htp</td>
<td>no</td>
</tr>
<tr>
<td>Old barnyard2 support</td>
<td>no</td>
</tr>
<tr>
<td>CUDA enabled</td>
<td>no</td>
</tr>
<tr>
<td>Hyperscan support</td>
<td>no</td>
</tr>
<tr>
<td>Libnet support</td>
<td>yes</td>
</tr>
<tr>
<td>Rust support (experimental)</td>
<td>no</td>
</tr>
<tr>
<td>Experimental Rust parsers</td>
<td>no</td>
</tr>
<tr>
<td>Rust strict mode</td>
<td>no</td>
</tr>
<tr>
<td>Suricatasc install</td>
<td>yes</td>
</tr>
<tr>
<td>Profiling enabled</td>
<td>no</td>
</tr>
<tr>
<td>Profiling locks enabled</td>
<td>no</td>
</tr>
</tbody>
</table>

**Development settings:**

- Coccinelle / spatch: yes
- Unit tests enabled: no
- Debug output enabled: no
- Debug validation enabled: no

**Generic build parameters:**

- Installation prefix: /usr
- Configuration directory: /etc/suricata/
- Log directory: /var/log/suricata/
- --prefix /usr
- --sysconfdir /etc
- --localstatedir /var

**Host:** x86_64-pc-linux-gnu

**Compiler:** gcc (exec name) / gcc (real)

**GCC Protect enabled:** no

**GCC march native enabled:** yes

**GCC Profile enabled:** no

**Position Independent Executable enabled:** no

**CFLAGS:** `-g -O2 -march=native`

**PCAP_CFLAGS:** `-I/usr/include`

**SECCFLAGS:**

**History**

#1 - 10/11/2017 01:55 PM - Peter Manev

- Subject changed from detect-thread-ratio has produces unexpected results with autofp to detect-thread-ratio produces unexpected results with autofp

#2 - 10/12/2017 04:40 PM - Andreas Herz

- Assignee set to OISF Dev
Seems to working as intended. There is 1 pcap receive thread, and then 8 and 16 'workers' as configured by the ratio. The ratio is applies as core/ht count * ratio. So this should be a box with 8 cores/hyperthreads.

Yes indeed - I forgot that 1 receive thread!
Wonder if it is worthed adding that to the output with autofp? (aka something like "all 1 pcap receive thread, 16 packet processing threads, 4 management threads initialized.")

Tracker changed from Bug to Feature
Subject changed from detect-thread-ratio produces unexpected results with autofp to more detailed output about number of threads created
Assignee changed from OISF Dev to Anonymous

Effort set to low
Difficulty set to low

Assignee set to Community Ticket