During my bypass testing, I've come across this memory leak.
I had set up Suricata with address sanitizers and an XDP bypass. I've used only two rules, which essentially should bypass all encountered traffic:

```
alert udp any any -> any any (msg: "UDP Packet!"; flow: to_server; bypass; sid: 999; rev:1;)
alert tcp any any -> any any (msg: "TCP Packet!"; flow: to_server; bypass; sid: 1000; rev:2;)
```

After transmitting some traffic against Suricata (to be very specific, it was 1M packets where Suricata detected 44724 flows) and then exiting, Asan has complained with following:

```
=================================================================
[52/6538]
==13985==ERROR: LeakSanitizer: detected memory leaks
Direct leak of 233576 byte(s) in 4171 object(s) allocated from:
 #0 0x7f072436eb77 in calloc (/lib64/libasan.so.6+0xb4b77)
 #1 0x979196 in SCCallocFunc /home/local/xsismi01/suricata/build/src/util-mem.c:57
 #2 0x9cbdae in PacketBypassCallback /home/local/xsismi01/suricata/build/src/decode.c:433
 #3 0xb75e08 in DetectBypassMatch /home/local/xsismi01/suricata/build/src/detect-bypass.c:96
 #4 0xb7d699 in DetectRunPostMatch /home/local/xsismi01/suricata/build/src/detect.c:177
 #5 0xb7f18c in DetectRulePacketRules /home/local/xsismi01/suricata/build/src/detect.c:805
 #6 0xb7d474 in DetectRun /home/local/xsismi01/suricata/build/src/detect.c:136
 #7 0xb81c43 in DetectFlow /home/local/xsismi01/suricata/build/src/detect.c:1683
 #8 0xb81ef5 in Detect /home/local/xsismi01/suricata/build/src/detect.c:1755
 #9 0xa8473c in FlowWorker /home/local/xsismi01/suricata/build/src/flow-worker.c:552
 #10 0x95979c in TMThreadsSlotVarRun /home/local/xsismi01/suricata/build/src/tm-threads.c:131
 #11 0xac3d85 in TMThreadsSlotProcessPkt /home/local/xsismi01/suricata/build/src/tm-threads.h:1
 #12 0xac5962 in AFPParsePacketV3 /home/local/xsismi01/suricata/build/src/source-af-packet.c:98
 #13 0xac5a69 in AFPPacketBlock /home/local/xsismi01/suricata/build/src/source-af-packet.c:996
 #14 0xac5bce in AFPPacketFromRingV3 /home/local/xsismi01/suricata/build/src/source-af-packet.c:1
 #15 0x95a098 in ReceiveAFLoop /home/local/xsismi01/suricata/build/src/source-af-packet.c:1393
 #16 0x95a098 in TMThreadsSlotPktAcqLoop /home/local/xsismi01/suricata/build/src/tm-threads.c:3
 #17 0x7f0722e90189 in start_thread (/lib64/libpthread.so.0+0x8189)
Indirect leak of 72 byte(s) in 3 object(s) allocated from:
 #0 0x7f072436eb77 in calloc (/lib64/libasan.so.6+0xb4b77)
```
Indirect leak of 48 byte(s) in 3 object(s) allocated from:
#0 0x7f072436eb77 in calloc (/lib64/libasan.so.6+0xb4b77)
#1 0x979196 in SCCallocFunc /home/local/xsisi01/suricata/build/src/util-mem.c:57
#2 0xac96d6 in AFPDPacketBypassCallback /home/local/xsisi01/suricata/build/src/source-af-packet.c:2351
#3 0xac96d6 in AFPDPacketBypassCallback /home/local/xsisi01/suricata/build/src/source-af-packet.c:2351

Indirect leak of 48 byte(s) in 3 object(s) allocated from:
#0 0x7f072436eb77 in calloc (/lib64/libasan.so.6+0xb4b77)
#1 0x979196 in SCCallocFunc /home/local/xsisi01/suricata/build/src/util-mem.c:57
#2 0xac96d6 in AFPDPacketBypassCallback /home/local/xsisi01/suricata/build/src/source-af-packet.c:2379

07/16/2022 2/3
I've tried to add the following snippet to `FlowQueuePrivateAppendFlow()` as I thought this function is used in (all) flow deletions. This however didn't help to solve the issue.

```c
if (f->flow_state == FLOW_STATE_CAPTURE_BYPASSED) {
    FlowBypassInfo *fc = FlowGetStorageById(f, GetFlowBypassInfoID());
    if (fc && fc->BypassFree && fc->bypass_data) {
        fc->BypassFree(fc->bypass_data);
        fc->bypass_data = NULL;
        fc->BypassFree = NULL;
    }
    SCFree(fc);
    FlowSetStorageById(f, GetFlowBypassInfoID(), NULL);
}
```

**Subtasks:**

Bug # 5422: bypass: Memory leak of some flow bypass objects. (6.0.x backport)  Closed

**History**

#1 - 05/20/2022 06:38 PM - Lukas Sismis

Probably a better function to use the snippet would be MoveToWorkQueue() but that did not help either.

I am using the following snippet because as per Asan output, decode.c on line 433 allocates bypass object but Asan thinks the object is not freed.

#2 - 05/21/2022 07:19 AM - Lukas Sismis

PR submitted.

#3 - 06/26/2022 05:57 PM - Victor Julien

- Status changed from New to Resolved
- Assignee changed from OISF Dev to Lukas Sismis
- Target version changed from TBD to 7.0rc1
- Label Needs backport to 6.0 added

https://github.com/OISF/suricata/pull/7418

#4 - 07/01/2022 10:42 AM - Victor Julien

- Status changed from Resolved to Closed

#5 - 07/01/2022 10:42 AM - Victor Julien

- Label deleted (Needs backport to 6.0)