Suricata - Bug #3099

Weird handling of IKEv2 flows when alerts happen

07/29/2019 05:32 PM - Antti Tönkyrä

**Status:** Closed  
**Priority:** Normal  
**Assignee:** Jason Ish  
**Category:**  
**Target version:** 6.0.0  
**Affected Versions:** 4.1.4  
**Difficulty:**  
**Effort:**  
**Label:**

**Description**

IKEV2 parser in its current state makes it possible to cause a slight denial-of-service condition to single suricata capture thread when alert is hit.

```plaintext
alert ikev2 any any -> ikev2.server.ip here any (msg:"SURICATA IKEv2"; sid:2; rev:1;)
```

My initial analysis follows.

It seems that the IKEv2 transaction vector for the flow in rust/src/ikev2/ikev2.rs (IKEV2State struct) gets filled with flows. Now let's say I fill it with 10000 IKEv2 exchanges that will amount to 10000 iterations (by get_tx_by_id) per new IKEv2 exchange and on top of that it seems that with every new IKEv2 exchange I make will cause get_tx_by_id be called 10 times. (so I get 100000 iterations per IKEv2 exchange which eats up fair bit of CPU time)

The flows are cleaned up as expected when flows timeout (which will clean up the transaction vector for that flow). Normally this would work but in this case the nature of IKE makes matters worse. Any new exchange will keep the flow alive (due to being srcip:500 <-> dstip:500) and this prevents the transaction vector from being cleaned up.

When alert does not happen, the connection is almost immediately cleaned up in AppLayerParserTransactionsCleanup (src/app-layer-parser.c) but the alert rule I described does cause us to hit goto next due to detect_flags_ts & APP_LAYER_TX_INSPECTED_FLAG being true. In non-alert case the transaction is freed by StateTransactionFree call before the next: goto label in AppLayerParserTransactionsCleanup function.

Tested on 4.1.4 and latest git head.

**History**

#1 - 07/29/2019 08:07 PM - Andreas Herz  
- Status changed from New to Assigned  
- Assignee set to OISF Dev  
- Target version set to 70  
- Private changed from No to Yes

Switched to private for now, Jason Ish can you take a look into that?

#2 - 08/07/2020 02:15 PM - Victor Julien  
- Target version changed from 70 to 6.0.0rc1

#3 - 09/09/2020 10:27 AM - Victor Julien  
- Target version changed from 6.0.0rc1 to 6.0.0

#4 - 10/05/2020 06:30 PM - Jason Ish  
I believe this to be the same issue as [https://redmine.openinfosecfoundation.org/issues/3877](https://redmine.openinfosecfoundation.org/issues/3877).

#5 - 10/06/2020 06:26 AM - Victor Julien  
- Status changed from Assigned to In Review
- Assignee changed from OISF Dev to Jason Ish

#6 - 10/07/2020 05:48 AM - Victor Julien
- Status changed from In Review to Closed

#7 - 12/04/2020 03:14 PM - Victor Julien
- Private changed from Yes to No