Suricata - Bug #3354

eve-log dns (possibly others) alerts miss metadata for all but first packet

11/21/2019 12:52 PM - Antti Tönkärä

| Status:   | Feedback           |
| Priority: | Normal             |
| Assignee: | Antti Tönkärä      |
| Category: |                    |
| Target version: | TBD     |
| Affected Versions: | 5.0.0   |
| Effort: |                    |

**Description**

It looks like eve-log alerts are slightly broken/miss metadata for DNS (might affect other protocols too).

I think it is caused by PacketAlertAppend(det_ctx, s, p, 0, alert_flags) call in DetectRulePacketRules which has argument of txid=0 to the appended alert. In my tests it seems to cause the effect that first DNS packet (query) has metadata on alert but response does not.

Ruleset I am using is simply:

```plaintext
alert dns any any -> any any (msg: "test"; sid: 1;)
```

And the produced EVE-json is

```json
{
    "timestamp": "2019-11-18T14:22:35.840789+0200",
    "flow_id": 1847130361418837,
    "pcap_cnt": 1,
    "event_type": "alert",
    "src_ip": "100.70.16.62",
    "src_port": 50073,
    "dest_ip": "1.0.0.1",
    "dest_port": 53,
    "proto": "UDP",
    "alert": {
        "action": "allowed",
        "gid": 1,
        "signature_id": 1,
        "rev": 0,
        "signature": "test",
        "category": "",
        "severity": 3
    },
    "dns": {
        "query": [
            {
                "type": "query",
                "id": 12920,
                "rrname": "b.fi",
                "rrtype": "A",
                "tx_id": 0
            }
        ],
        "app_proto": "dns",
        "flow": {
            "pkts_toserver": 1,
            "pkts_toclient": 0,
            "bytes_toserver": 64,
            "bytes_toclient": 0
        }
    }
}
```
"bytes_toclient": 0,
"start": "2019-11-18T14:22:35.840789+0200",
"payload": "MngBAAAABAAAAAAAAAWICZmkAAAEAAQ==",
"stream": 0
}
}
"timestamp": "2019-11-18T14:22:35.856384+0200",
"flow_id": 1847130361418837,
"pcap_cnt": 2,
"event_type": "alert",
"src_ip": "1.0.0.1",
"src_port": 53,
"dest_ip": "100.70.16.62",
"dest_port": 50073,
"proto": "UDP",
"alert": {
  "action": "allowed",
  "gid": 1,
  "signature_id": 1,
  "rev": 0,
  "signature": "test",
  "category": "",
  "severity": 3
},
"app_proto": "dns",
"flow": {
  "pkts_toserver": 1,
  "pkts_toclient": 1,
  "bytes_toserver": 64,
  "bytes_toclient": 80,
  "start": "2019-11-18T14:22:35.840789+0200"
},
"payload": "MniBgAABAAEAAAAAWICZmkAABAAEAAAAAAMkAAAEAAQ==",
"stream": 0
}

History
#1 - 11/23/2019 11:05 PM - Andreas Herz
- Assignee set to OISF Dev
- Target version set to TBD

#2 - 11/24/2019 06:28 AM - Victor Julien
- Status changed from New to Feedback
- Assignee changed from OISF Dev to Antti Tönkyrä

Can you submit a Suricata-Verify test for this case?

#3 - 11/24/2019 11:03 AM - Antti Tönkyrä
Sure, [https://github.com/OISF/suricata-verify/pull/160](https://github.com/OISF/suricata-verify/pull/160)

#4 - 12/18/2019 02:00 PM - Antti Tönkyrä
- File flowtx.patch added

I'm not sure if the correct fix would be to add looking up TX like we do in stateless sigs at [https://github.com/OISF/suricata/blob/3887f8d1f3d2816b3f46fb48560f9de57ae66314/src/detect.c#L1292](https://github.com/OISF/suricata/blob/3887f8d1f3d2816b3f46fb48560f9de57ae66314/src/detect.c#L1292)
to [https://github.com/OISF/suricata/blob/3887f8d1f3d2816b3f46fb48560f9de57ae66314/src/detect.c#L807](https://github.com/OISF/suricata/blob/3887f8d1f3d2816b3f46fb48560f9de57ae66314/src/detect.c#L807)

The necessary information to call the function GetDetectTx to obtain the correct TX seems to be there and I made a quick hack which seemed to fix the issue (patch as attachment). I wonder if this is the correct way to fix the issue? If so I can do a PR with some cleanup done.

Files