### Suricata - Bug #4376

**TCP flow that retransmits the SYN with a newer TSval not properly tracked**

03/02/2021 04:53 PM - Nate Shimp

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Michael Tremer</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>7.0rc1</td>
</tr>
<tr>
<td>Affected Versions:</td>
<td>5.0.3</td>
</tr>
<tr>
<td>Effort:</td>
<td></td>
</tr>
<tr>
<td>Difficulty:</td>
<td></td>
</tr>
<tr>
<td>Label:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

We've run into a situation where Suricata 5.0.3 ignores a TCP flow because the 3-way handshake starts with:

```
SYN (TSval=X, TSecr=0) -->
SYN (retransmitted, TSval=Y, TSecr=0) -->
<-- SYN, ACK (TSval=Z, TSecr=Y)
```

The TCP state machine only stores the original SYN's TSval in the session, so `StateSynSentValidateTimestamp()` says the timestamps are invalid when it sees the SYN-ACK.

We have a fix for this that updates the session's timestamp if Suricata sees another SYN with a newer timestamp:

```diff
diff --git a/src/stream-tcp.c b/src/stream-tcp.c
index a99ce68b6..8e218f460 100644
--- a/src/stream-tcp.c
+++ b/src/stream-tcp.c
@@ -1641,6 +1641,21 @@
        } else if (PKT_IS_TOSERVER(p)) {
            /* This covers a subtle edge case where a SYN is retransmitted with a newer
                * TSval. If the the session's last_ts value isn't updated, the SYN-ACK
                * will fail timestamp validation and the flow will be lost.
                */
            + if ((ssn->client.flags & STREAMTCP_STREAM_FLAG_TIMESTAMP) != 0) {
                + SCLogDebug("SYN retransmitted with timestamp options on packet ", ssn
                        + "\$PRIu64", ssn, 0x016); if (TCP_HAS_TS(p)) {
                    + uint32_t ts_val = TCP_GET_TSVAL(p);
                    + if (ssn->client.last_ts < ts_val) {
                        + ssn->client.last_ts = ts_val;
                        + ssn->client.last_pkt_ts = p->ts.tv_sec;
                    + }
                + }
            + }
        + }
        +
    /** \todo check if it's correct or set event */
```

We're wondering if this is a reasonable fix, or if upheaving some security precaution.

**Subtasks:**

- Bug #5421: TCP flow that retransmits the SYN with a newer TSval not properly tracked (Closed)
- Bug #5429: TCP flow that retransmits the SYN with a newer TSval not properly tracked (Closed)

**History**

#1 - 03/02/2021 07:05 PM - Victor Julien
- Status changed from New to Assigned
- Assignee set to Victor Julien

07/14/2022
Thanks Nate, I will have a look.

#2 - 06/10/2022 12:26 PM - Victor Julien
- Subject changed from Suricata ignores TCP flow that retransmits the SYN with a newer TSval to TCP flow that retransmits the SYN with a newer TSval not properly tracked
- Status changed from Assigned to In Review

https://github.com/OISF/suricata/pull/7523

#3 - 06/10/2022 12:27 PM - Victor Julien
- Assignee changed from Victor Julien to Michael Tremer

#4 - 06/10/2022 12:30 PM - Victor Julien
- Label Needs backport to 5.0, Needs backport to 6.0 added

#5 - 07/01/2022 09:29 AM - Victor Julien
- Status changed from In Review to Resolved

https://github.com/OISF/suricata/pull/7538/commits/117278cb77d9015a1014316dc055c20bfdd653c

#6 - 07/01/2022 10:52 AM - Victor Julien
- Label deleted (Needs backport to 6.0)

#7 - 07/01/2022 10:56 AM - Victor Julien
- Status changed from Resolved to Closed
- Label deleted (Needs backport to 5.0)

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>syn_retransmit_with_ts.pcap</td>
<td>228 KB</td>
<td>03/02/2021</td>
<td>Nate Shimp</td>
</tr>
</tbody>
</table>