Hi,

packet retransmissions from client to server seem to be dropped by Suricata if client has previously sent a FIN packet to server. This could be why our FTP uploads get stuck when network load is high.

In our tests, we are uploading several thousands of 8 KB files from client (192.168.2.2) to server (192.168.3.3). You can see in the attached files a report with the packets of a stuck FTP-DATA connection as captured from the client and from the server. Also, I have attached the suricata debug output belonging to that connection.

Packet No. 43992 in the client-side capture was not received by the server. Client sent finally a FIN and several retransmissions of the lost packet latter, but Suricata might have dropped them because of its transition to CLOSE_WAIT state.

Suricata build info:

This is Suricata version 2.0 RELEASE
Features: DEBUG NFQ PCAP_SET_BUFF LIBPCAP_VERSION_MAJOR=1 AF_PACKET HAVE_PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAVE_HTP_URI_NORMALIZE_HOOK PCRE_JIT SIMD support: none
Atomic intrinsics: 1 2 4 8 byte(s)
64-bits, Little-endian architecture
GCC version 4.7.2, C version 199901
L1 cache line size (CLS)=64
compiled with LibHTP v0.5.10, linked against LibHTP v0.5.10
Suricata Configuration:

AF_PACKET support: yes
PF_RING support: no
NFQueue support: yes
IPFW support: no
DAG enabled: no
Napatech enabled: no
Unix socket enabled: no
Detection enabled: yes

libnss support: no
libnspr support: no
libjansson support: no
Prelude support: no
PCRE jit: yes
libluajit: no
libgeoip: no
Non-bundled htp: no
Old barnyard2 support: no
CUDA enabled: no

Suricatasc install: yes

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

Generic build parameters:
Installation prefix (--prefix): /opt/suricata-2.0debug
Configuration directory (--sysconfdir): /etc/suricata/
Log directory (--localstatedir): /var/local/suricata-2.0debug/log/suricata/

Host: x86_64-unknown-linux-gnu
GCC binary: gcc
GCC Protect enabled: no
GCC march native enabled: no
GCC Profile enabled: no

History

#1 - 04/15/2014 12:09 PM - Anoop Saldanha
I'm probably wondering if this is because the tcp state transitions for both client and server are maintained in one variable, which will lead to wrong transitions, while the var should either reflect either the client or the server state.

#2 - 04/16/2014 03:35 AM - Victor Julien
- Description updated
- Status changed from New to Assigned
- Assignee set to Victor Julien
- Target version set to 2.0.1rc1

#3 - 04/16/2014 03:41 AM - Victor Julien
I think I see the issue. The problem seems to be that StreamTcpPacketIsRetransmission() returns 2 different return codes for 2 different cases. It detects retransmissions of already ack'd data (which is thus received by the server) and of un-ack'd data (which may be lost after Suricata saw it). Currently we just flat out reject retransmissions in the CLOSE_WAIT state.

Maybe you can try changing line stream-tcp.c:3597 from

```c
if (StreamTcpPacketIsRetransmission(&ssn->client, p)) {
```

to

```c
if (StreamTcpPacketIsRetransmission(&ssn->client, p) == 2) {
```

And report back if that solves this case.

If you can (privately) share a pcap it would be great.

#4 - 04/16/2014 07:46 AM - Victor Julien
Can you try the changes here? [https://github.com/inliniac/suricata/pull/935](https://github.com/inliniac/suricata/pull/935)

#5 - 04/17/2014 08:51 AM - Victor Julien
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

Merged [https://github.com/inliniac/suricata/pull/941](https://github.com/inliniac/suricata/pull/941)

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-capture.txt</td>
<td>3.27 KB</td>
<td>04/15/2014</td>
<td>Gines Lopez</td>
</tr>
<tr>
<td>server-capture.txt</td>
<td>2.28 KB</td>
<td>04/15/2014</td>
<td>Gines Lopez</td>
</tr>
<tr>
<td>debug.txt</td>
<td>22 KB</td>
<td>04/15/2014</td>
<td>Gines Lopez</td>
</tr>
</tbody>
</table>