**Suricata - Feature #3440**

Task # 4966 (New): tracking: QUIC protocol support

**Add GQUIC Protocol Analysis and CYU Fingerprinting**

01/15/2020 04:31 PM - John Althouse

<table>
<thead>
<tr>
<th>Status:</th>
<th>In Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Emmanuel Thompson</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>7.0rc1</td>
</tr>
<tr>
<td>Effort:</td>
<td></td>
</tr>
<tr>
<td>Difficulty:</td>
<td></td>
</tr>
<tr>
<td>Label:</td>
<td>Protocol</td>
</tr>
</tbody>
</table>

**Description**

QUIC traffic could be up to 20% of an organizations network traffic if they use Google products, like Chrome. Background: https://engineering.salesforce.com/gquic-protocol-analysis-and-fingerprinting-in-zeek-a4178855d75f

Here's the protocol analyzer and fingerprinting script for Zeek for reference: https://github.com/salesforce/GQUIC_Protocol_Analyzer

**Related issues:**

Precedes Feature #4967: QUIC v1 support In Review

---

**History**

**#1 - 07/13/2020 08:32 PM - Emmanuel Thompson**

I can look into this. Here's what I'm thinking:

**QUIC AppLayer**

Parser: Rust
- Look at using https://github.com/cloudflare/quiche, downside is: quiche requires Rust 1.39 or later to build.
- Write a nom parser, using nom-derive
- Long and short header versions

**Keyword:**
- CYU (much like JA3 but for QUIC):
  - Generate CYU hash and tags for the QUIC ClientHello packet
  - Create a ‘cyu.string’ and ‘cyu.hash’ keyword

**Logging:**
- Anything required here?

**Other resources:**

Quiche parsing: https://docs.rs/quiche/0.3.0/src/quiche/packet.rs.html#173

Are there other areas which should be looked at?

**#2 - 07/14/2020 09:39 AM - Victor Julien**

- Status changed from New to Assigned
- Assignee set to Emmanuel Thompson
- Target version set to 7.0rc1

The rustc 1.39 requirement is a problem. Have you looked at ‘quiche’ in general to see how well it would fit our parsing and state keeping model?

Wrt logging: I'd say we'd want a logger for at least the CYU hash and string.

**#3 - 07/14/2020 09:39 AM - Victor Julien**

- Label Protocol added
Quiche could be nice for parsing, we can then manipulate on the parsed values. It also has support for many drafts of the protocol (as it's not stabilized yet)

Latest draft: https://tools.ietf.org/html/draft-ietf-quic-transport-29#section-17

I'll implement a parser for it.

FYI GQUIC is a predecessor of QUIC IETF

The parser in the OP parses GQUIC Versions Q039-Q046.

@John Althouse, would you have specific PCAPs for testing?

Let me add in Caleb Yu here.

Here are two PCAPs with GQUIC traffic from my Linux VM:
The first one should have a CYU value of a46560d4548108cf99308319b3b85346 from the version and tags
The second one has two client hellos:
1. 910a5e3a4d51593bd59a446115441209 from
2. 7b3ceb1adc974ad360cfa634e8d0a730 from

https://github.com/OISF/suricata/pull/5556

- Subject changed from Add QUIC Protocol Analysis and CYU Fingerprinting to Add GQUIC Protocol Analysis and CYU Fingerprinting
- Parent task set to #4966

- Precedes Feature #4967: QUIC v1 support added

<table>
<thead>
<tr>
<th>Files</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>GQUIC1.pcapng</td>
<td>7.93 KB</td>
<td>09/02/2020</td>
<td>Caleb Yu</td>
</tr>
<tr>
<td>GQUIC2.pcapng</td>
<td>9.84 KB</td>
<td>09/02/2020</td>
<td>Caleb Yu</td>
</tr>
</tbody>
</table>