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>Closed</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>
</tbody>
</table>

**Label: Protocol**

**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

**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

07/17/2022
#4 - 07/15/2020 01:49 PM - Emmanuel Thompson
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.

#5 - 07/16/2020 01:08 PM - Emmanuel Thompson
FYI GQUIC is a predecessor of QUIC IETF

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

#6 - 08/31/2020 06:02 PM - Emmanuel Thompson
@John Althouse, would you have specific PCAPs for testing?

#7 - 09/02/2020 02:47 PM - John Althouse
Let me add in Caleb Yu here.

#8 - 09/02/2020 04:22 PM - Caleb Yu
- File GQUIC1.pcapng added
- File GQUIC2.pcapng added

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

#9 - 11/12/2020 11:23 AM - Victor Julien
- Status changed from Assigned to In Review

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

#10 - 01/17/2022 07:07 AM - Victor Julien
- Subject changed from Add QUIC Protocol Analysis and CYU Fingerprinting to Add GQUIC Protocol Analysis and CYU Fingerprinting
- Parent task set to #4966

#11 - 01/17/2022 07:28 AM - Victor Julien
- Precedes Feature #4967: QUIC v1 support added

#12 - 03/04/2022 11:40 AM - Victor Julien
- Status changed from In Review to Closed

https://github.com/OISF/suricata/pull/6813