Suricata - Feature #3260
SMTP Base64 Decoding of Message Body
10/17/2019 06:47 PM - Brandon Murphy

<table>
<thead>
<tr>
<th>Status:</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>OISF Dev</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>TBD</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**

In attempting to write signatures for the Base64 decoded message body (not an attachment) I find that signatures do not fire as expected.

FWIW, I’ve observed the same issue for quoted-printable encoding of the message body as well. But intend on opening another issue for that.

I have attached a zip that contains pcap, IDS output, rules, etc.

The pcap has two different tcp sessions, one which includes only a base64 message body (tcp.port 4204), and another that includes the same base64 as an attachment (tcp.port 44228). This was done to validate that base64 decoding of SMTP attachments worked and could be triggered via the file_data keyword as per [https://redmine.openinfosecfoundation.org/issues/3190](https://redmine.openinfosecfoundation.org/issues/3190).

The pcap is capturing while using python scripts to read an EML and send it to a python "SMTP Sink". I then used tcprewrite to rewrite the src/dst ips from 127.0.0.1 to something more representative of the traffic I'm attempting to signature.

Please let me know if anything else is required.

**Related issues:**
- Related to Feature #3261: SMTP quoted-printable Decoding of Message Body
  - New
- Related to Task #4097: Suricon 2020 brainstorm
  - New

**History**

#1 - 10/20/2019 08:55 PM - Andreas Herz
- Assignee set to OISF Dev
- Target version set to TBD

#2 - 10/21/2019 01:57 PM - Victor Julien
- Tracker changed from Bug to Feature

I think this would require new keywords.

When looking at the example rules:

```
alert smtp any any -> any any (msg:"base64 message body test 1"; content:"|22|Elephants|22|"; sid:1; rev:1; classtype:trojan-activity;)
alert smtp any any -> any any (msg:"base64 message body test 2 includes file_data"; file_data; content:"|22|Elephants|22|"; sid:2; rev:1; classtype:trojan-activity;)
```

Sid:1 inspects the raw stream. There is no overloading of this to actually replace with some other buffer, by design.

Sid:2 inspects file_data. This does not include the message, just attachments.

So I think we'd need something like a 'smtp.body' sticky buffer for this, that would indeed base64 decode as needed.

#3 - 10/21/2019 01:58 PM - Victor Julien
- Related to Feature #3261: SMTP quoted-printable Decoding of Message Body added
#4 - 01/30/2020 11:54 AM - Victor Julien
- Label Protocol added

#5 - 11/14/2020 02:26 PM - Jeff Lucovsky
- Related to Task #4097: Suricon 2020 brainstorm added

Files
<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>base64_message_body_trbl.zip</td>
<td>32.6 KB</td>
<td>10/17/2019</td>
<td>Brandon Murphy</td>
</tr>
</tbody>
</table>