As reported on oisf-devel:

Observed 2 segmentation faults while using Suricata. The traffic used for the test was a (large) merged pcap of multiple pcaps available on pcapr.net.

Although that pcap cannot be retrieved and shared, I have noted some preliminary GDB analysis that might help identify the issues.

1) Segmentation fault occurred at "libhttp/htp_response_generic.c" 279L
   Code: size_t len = bstr_len(tx->response_line);
   (gdb) p tx->response_line
   $24 = (bstr *) 0x0
   (gdb) p tx->response_line.ptr
   Cannot access memory at 0x0
   Macro bstr_len tried to dereference a NULL pointer (response_line)

2) Segmentation fault occurred at "src/detect-http-method.c" 697L
   Code: for(idx = 0; idx < list_size(hs->connp->conn->transactions); idx++)
   The pointer "hs" is NULL, and is being dereferenced. Suricata does have a check to detect whether this ptr is NULL.
   However, the check is performed before acquiring a semaphore. Apparently, things change by the time the semaphore is acquired.
   Perhaps, the checks need to be performed before and after the semaphore operation.

Check the not NULL of connp and conn before using them. I guess that the problem is more related to libhtp, that unset the pointer connp, but anyway I think that the engine must check this pointers to avoid a crash, so uploading a patch for this.

It would be nice to have a pcap of that filtered session, and/or suricata logs to know how that pointer ended to be NULL.

That patch has a strange character and it doesn't compile correctly. Use this one instead.
Btw, what about the fix at libhtp? should I send a patch? or ping Ivan?

That second segv is quite strange. It is true that hs is set to "state" before the lock, but it's a local variable. So even if the htp state changes, hs would never change. It could point to invalid memory, but the var itself should never become NULL unless state was NULL when hs was set. But then the if (hs == NULL) check should have fixed that. Did you compile suricata at a high optimization level perhaps? Maybe gcc reordered code so the check is bypassed...
#4 - 08/30/2010 05:04 AM - Pablo Rincon
I don't think that the null pointer is hs. I think that connp was NULL. Can you please rerun that pcap with gdb and attach here the backtrace, also printing hs and hs->connp and hs->connp->conn? It would be helpful.

#5 - 08/30/2010 06:17 AM - Victor Julien
I've applied your patch Pablo.

Let's report the htp issue to Ivan if we have more detailed info like a backtrace.

#6 - 09/02/2010 04:27 AM - Victor Julien
- Target version set to 1.0.2

Closing for now. Please reopen or open a new ticket if the issue resurfaces.

#7 - 09/02/2010 04:27 AM - Victor Julien
- Status changed from Assigned to Closed
- % Done changed from 50 to 100

Files

<table>
<thead>
<tr>
<th>Patch Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001-Fix-a-segv-if-there-is-an-error-at-libhtp-no-connp.patch</td>
<td>792 Bytes</td>
<td>08/27/2010</td>
<td>Pablo Rincon</td>
</tr>
<tr>
<td>0001-Fix-segv-condition-on-DetectHttpMethodMatch-if-the-a.patch</td>
<td>816 Bytes</td>
<td>08/27/2010</td>
<td>Pablo Rincon</td>
</tr>
</tbody>
</table>