Security Assessment of the Transmission Control Protocol (TCP)
(draft-ietf-tcpm-tcp-security-02.txt)

Fernando Gont

project carried out on behalf of
UK CPNI

80th IETF meeting, Prague, Czech Republic
March 27-April 1, 2011
Working Process

- At the Anaheim IETF, a process was agreed upon to evaluate the recommendations in this document.

- The process aims to categorize each recommendation as:
  - Implementation issues
  - Operational issues
  - Wiggle room in the specification
  - Bug in the document
  - Bug in the specification

- For each category, there is a clear way forward

- The process can be summarized with a set of questions.
Process flow “chart”

- Do we agree X is correct?
  - No: Bug in the document – remove.
  - Yes: CONTINUE

- Implementation issue?
  - Yes: Document (as updated to RFC 2525)
  - No: CONTINUE

- Operational (config) issue?
  - Yes: Is this a good default?
    - Yes: Recommend default config
    - No: Discuss as config option
  - No: CONTINUE
Process flow “chart” (cont.)

- Wiggle room in the specification?
  - Yes: Discuss as valid exception between MAY/SHOULD
  - No: Does this warrant adding wiggle room?
    - Yes: Downgrade MUST to SHOULD
    - No: CONTINUE

- Change the spec
Current version of the document

- TCPM began to review some recommendations on the mailing list and in Anaheim, but had difficulty since recommendations weren't clearly identified from rationale.
- As agreed in Beijing IETF, version -02 is organized in RFC1122-style: recommendations are now more easily identified.
- Much text was replaced with references to existing RFCs (more to come in this area).
- Reviews are highly needed (a few people have signed up, already).
## Summary of recommendations

<table>
<thead>
<tr>
<th>Section</th>
<th># Recs</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Header Fields</td>
<td>23</td>
</tr>
<tr>
<td>4. TCP Options</td>
<td>18</td>
</tr>
<tr>
<td>5. Connection Establishment</td>
<td>8</td>
</tr>
<tr>
<td>6. Connection Termination</td>
<td>1</td>
</tr>
<tr>
<td>7. Buffer Management</td>
<td>3</td>
</tr>
<tr>
<td>8. Segment Reassembly</td>
<td>1</td>
</tr>
<tr>
<td>9. Congestion Control</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th># Recs</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. TCP API</td>
<td>4</td>
</tr>
<tr>
<td>11. Blind In-window attacks</td>
<td>5</td>
</tr>
<tr>
<td>12. Information Leaking</td>
<td>5</td>
</tr>
<tr>
<td>13. Covert Channels</td>
<td>0</td>
</tr>
<tr>
<td>14. TCP Port scanning</td>
<td>3</td>
</tr>
<tr>
<td>15. TCP processing of ICMP</td>
<td>3</td>
</tr>
<tr>
<td>16. TCP and IP Interaction</td>
<td>1</td>
</tr>
</tbody>
</table>
Technical Discussion
Acknowledgement number check

- The Acknowledgement Number was required to be:
  - SEG.ACK $\leq$ SND.NXT

- RFC 5961 [Ramaiah et al, 2010] proposed a stricter check:
  - SND.UNA - SND.MAX.WND $\leq$ SEG.ACK $\leq$ SND.NXT
  - If a segment does not pass this check, it should be dropped.

- Specification issue:
  - *TCP MUST* check that, on segments that have the ACK bit set, the Acknowledgment Number satisfies the expression: SND.UNA - SND.MAX.WND $\leq$ SEG.ACK $\leq$ SND.NXT
  - If a TCP segment does not pass this check, the segment *MUST* be dropped, and an ACK segment *SHOULD* be sent in response.
Acknowledgement number

- Some stacks fail to set the Acknowledgement Number to zero when the ACK bit is not set (e.g., SYN segments or RST segments)
- This may produce an information leakage
- Implementation issue:
  - TCP SHOULD set the Acknowledgement Number to zero when sending a TCP segment that does not have the ACK bit set (i.e., a SYN segment).
Urgent Pointer

Basic Principle:
- TCP MUST check that: Segment.Size - Data Offset * 4 > 0
- If a TCP segment with the URG bit set does not pass this check, it MUST be silently dropped.

Implementation issue:
- For TCP segments that have the URG bit set to zero, sending the TCP SHOULD set the Urgent Pointer to zero.

Basic Principle:
- A receiving TCP MUST ignore the Urgent Pointer field of TCP segments for which the URG bit is zero.