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

Section	# Recs
3. Header Fields	23
4. TCP Options	18
5. Connection Establishment	8
6. Connection Termination	1
7. Buffer Management	3
8. Segment Reassembly	1
9. Congestion Control	7

Section	# Recs
10. TCP API	4
11. Blind In-window attacks	5
12. Information Leaking	5
13. Covert Channels	0
14. TCP Port scanning	3
15. TCP processing of ICMP	3
16. TCP and IP Interaction	1

Technical Discussion

Acknowledgement number check

- The Acknowledgement Number was required to be:
  - SEG.ACK <= SND.NXT
- RFC 5961 [Ramaiah et al, 2010] proposed a stricter check:
  - SND.UNA - SND.MAX.WND <= SEG.ACK <= 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 <= SEG.ACK <= 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 leakege
- 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.
- Implemetation 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.