ICMP attacks against TCP
(draft-gont-tcpm-icmp-attacks-05.txt)

Fernando Gont

UTN/FRH

64th IETF meeting, Vancouver, BC, Canada

November 6-11, 2005

Overview

- ICMP can be used to perform a number of attacks against TCP, which include:
  - Blind connection-reset attacks
  - Blind throughput-reduction attacks
  - Blind performance-degrading attacks
- In April 2005, UK's NISCC, US-CERT, and most major vendors (including Cisco, Microsoft, IBM, Juniper, RedHat, Sun, and HP, among others) published vulnerability reports on these vulnerabilities.
- A large number of implementations, ranging from desktop systems to core Internet routers, were found vulnerable to either all or a subset of these attacks.
- **draft-gont-tcpm-icmp-attacks** proposes counter-measures for these attacks. It has benefited from the insights of the TCPM WG, the PMTUD WG, Sun Microsystems, the FreeBSD, NetBSD, OpenBSD, and Linux projects, and other professionals.

Considerations

- In order to perform these attacks, all an attacker needs to know/guess is the four-tuple {local IP address, local TCP port, remote IP address, remote TCP port} that identifies the TCP connection to be attacked.
- Thus, ICMP-based attacks are much easier to perform than TCP-based ones (even no need to “hit the window”!)
- These attacks don’t depend on source IP address spoofing. Simple ingress/egress filtering does not help to mitigate them.
- It is important to note that ICMP messages are **unreliable**. Therefore, if some ICMP message is dropped due to performing validation checks, interoperativity won’t be affected. (After all, the ICMP message could have been lost due to corruption, congestion, and/or rate-limiting!).
- Furthermore, in many cases, the ICMP error messages could have been elicited by **corrupted segments**!
- As indicated by **draft-iab-link-indications-03.txt,**

  “*Given today's security environment, it is inadvisable for hosts to act on indications provided by gateways without careful consideration.*”

Mitigating the blind connection-reset attack

- **From D. D. Clark's "Fault Isolation and Recovery" (RFC 816):**

"***To abandon a TCP connection based on such a message arriving would be to ignore the valuable feature of the Internet that for many internal failures it reconstructs its function without any disruption of the end points***. *But if failure messages do not imply a failure, what are they for? In fact, error messages serve several important purposes.*" "*….they provide valuable information, after the TCP timeout has occurred, as to the probable cause of the failure.*"

"*....**In general, error messages give valuable information about what went wrong, but are not to be taken as absolutely reliable.** A general alerting mechanism, such as the TCP timeout discussed above, provides a good indication that whatever is wrong is a serious condition, but without the advisory messages to augment the timer, there is no way for the client to know how to respond to the error. **The combination of the timer and the advice from the error messages provide a reasonable set of facts for the client layer to have.***"

For connections in any of the synchronized states, treat the so-called "hard errors" as "soft errors", unless the TCP segment contained in the ICMP payload has a valid checksum, is in-window, and has a correct TCP MD5 signature.

Mitigating the blind throughput-reduction attack

- Use of ICMP Source Quench messages for congestion control has been deprecated for quite a while, even in the specifications themselves (RFC 1812). TCP does not use them for flow-control, either.
- However, TCP is still required to slow down the rate at which it is sending information if ICMP Source Quench messages are received (RFC 1122).

Therefore, ignore ICMP Source Quench messages meant for TCP connections.

Mitigating the blind performance-degrading attack

- Divide PMTUD into two phases: Initial PMTUD, and PMTU Update.
- The Initial PMTUD phase is when we have no records of “large” packets getting to the remote end-point (most likely, the connection has just been established). In this phase, perform the traditional PMTUD.
- The PMTU Update phase is when the network asks us to reduce the size of the packets we send. In this case, we must be more cautious, as we have records of “large” packets getting to the remote endpoint. Therefore, record the received ICMP message, and wait for at least one RTO. If the TCP segment contained in the ICMP payload gets acknowledged, disregard the error message. If not, honor it, updating the Path-MTU.
- Currently cooperating with the PMTUD WG to integrate the proposed ICMP processing with PLPMTUD. (A big thank you to Matt Mathis and John Heffner for their insights, by the way!)

(Extended explanation, sample scenarios, pseudo-code, etc., available in the draft. Also, there’s some pending feedback to be included.)

Running code

(Implementation of the proposed counter-measures)

Implemented counter-measures	Connection reset	Throughput reduction	performance degrading
Linux	**Yes**	**Yes**	**Upcoming**
FreeBSD	**Yes**	**Yes**	**Upcoming**
NetBSD	**Yes**	**Yes**	**Yes**
OpenBSD	**Yes**	**Yes**	**Yes**
Solaris	**Yes**	**Yes**	**Partially**

- All BSD-derived and Mentat-derived TCP/IP implementations have traditionally implemented the proposed processing of the so-called ICMP “hard errors” (for more than 15 years).
- Most implementations have removed support for ICMP Source Quench messages meant for TCP connections since NISCC’s disclosure.
- Virtually every implementation now checks, at least, the TCP SEQ number contained in the ICMP payload.

Issues raised on the mailing-list

- “The document should not focus on security. It should focus on ICMP messages caused by stale segments, and mention that attacks are addressed as a corollary”

Author’s point of view:

- None of the existing counter-measures are based on the concept of “stale segments”. The ones for the connection-reset and throughput-reduction attacks are based on a change in the processing of the respective ICMP error messages. The counter-measure for the PMTUD attack is based on checking progress, not staleness.
- Addressing ICMP error messages caused by stale segments does not address the possible attacks. TCP addresses stale TCP segments. Guess why we are working on tcp-secure.

Actually, the **converse** is true. By addressing attacks, we also handle ICMP messages elicited by stale and/or corrupted segments, etc.

(This could be explicited in an appendix, though.)

Moving forward

- The draft is the result of the collaborative work of the open source community, commercial vendors, and the IETF.
- The industry has adopted the proposed counter-measures, and has referenced the draft in their vulnerability reports.
- The draft has been referenced in **draft-iab-link-indications-03.txt** as a proposal on how to deal with ICMP attacks against TCP.

Should we take the draft as a WG item?

Feedback?

Fernando Gont

fernando@gont.com.ar

http://www.gont.com.ar