ICMP attacks against TCP
draft-gont-tcpm-icmp-attacks-01.txt

Status of this Memo

Copyright Notice

Abstract

   This document discusses the use of the Internet Control Message
   Protocol (ICMP) to perform a variety of attacks against the
   Transmission Control Protocol (TCP) and other similar protocols. It
   proposes a work-around to eliminate or minimize the impact of this
   type of attack.

Table of Contents

1. Introduction

Recently, awareness has been raised about several threats against the
TCP [1] protocol, which include blind connection-reset attacks [6].
These attacks are based on sending forged TCP segments to any of the
TCP endpoints, requiring the attacker to be able to guess the
four-tuple that identifies the connection to be attacked.

While these attacks were known by the research community, they were
considered to be unfeasible. However, increases in bandwidth
availability, and the use of larger TCP windows [7] have made these
attacks feasible. Several solutions have been proposed to either
eliminate or minimize the impact of these attacks [8][9][10].

However, there is still a possibility for performing a number of
attacks against the TCP protocol, by means of ICMP [2]. These
attacks include, among others, blind connection-reset attacks.

This document aims to raise awareness of the use of ICMP to perform a
number of attacks against TCP, and proposes several counter-measures
that can eliminate or minimize the impact of these attacks.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [3].

2. Background

2.1 Internet Control Message Protocol (ICMP)

The Internet Control Message Protocol (ICMP) is used by the Internet
Architecture to perform the fault-isolation function, that is, the
group of actions that hosts and routers take to determine that there
is some network failure [11].

When an intermediate router detects a network problem while trying to
forward an IP packet, it will send an ICMP error message to the
source host, to raise awareness of the network problem. In the same
way, there are a number of cases in which an end-system may generate
an ICMP error message when it finds a problem while processing a
datagram.

The internet header plus the first 64 bits of the packet that
elicited the ICMP message are included in the payload of the ICMP
error message, so that the receiving host can match the error to the
instance of the transport protocol that elicited the error message.
Thus, it is assumed that all data needed to identify a transport
protocol instance is contained in the first 64 bits of the transport

protocol header.

When the transport protocol is notified of the error condition, it
will perform a fault recovery function. That is, it will try to
survive the network failure.

In the case of TCP, the fault recovery policy is as follows:

o If the network problem being reported is a hard error, abort the
   corresponding connection.

o If the network problem being reported is a soft error, just record
   this information, and repeatedly retransmit the segment until
   either it gets acknowledged, or the connection times out.

The Host Requirements RFC [4] defines "hard errors" as ICMP error
messages of type 3 (Destination Unreachable) codes 2 (protocol
unreachable), 3 (port unreachable), and 4 (fragmentation needed and
DF bit set). Thus, any of these ICMP messages could elicit a
connection abort.

[12] provides information about which ICMP error messages are
produced by hosts, intermediate routers, or both.

2.2 Handling of ICMP errors

The Host Requirements RFC [4] states that a TCP instance should be
notified of ICMP error messages received for its corresponding
connection. However, neither the Host Requirements RFC nor the
original TCP specification [1] recommend any additional security
checks on the received ICMP messages.

Therefore, as long as the ICMP payload contains the correct
four-tuple that identifies the communication instance, it will be
processed by the corresponding transport-protocol instance, and the
corresponding action will be performed.

Thus, in order to perform any of the attacks discussed in this
document, an attacker only needs to guess the four-tuple that
identifies the communication instance to be attacked. As discussed
in [6], there are a number of scenarios in which an attacker may be
able to know or guess this four-tuple.

Furthermore, it must be noted that most services use the so-called
"well-known" ports, so that only the client port would need to be
guessed. In the event that an attacker had no knowledge about the
range of port numbers used by clients, this would mean that an
attacker would need to send, at most, 65535 packets to perform any of

the attacks described in this document.

It is clear that additional security checks should be performed on the received ICMP error messages.

[3](#). ICMP attacks against TCP

ICMP messages can be used to perform a variety of attacks. These attacks have been discussed by the research community to a large extent.

Some TCP/IP implementations have added extra security checks on the received ICMP error messages to minimize the impact of these attacks. However, as there has not been any official proposal about what would be the best way to deal with these attacks, these additional security checks have not been widely implemented.

The following subsections discuss some of the possible attacks, and propose work-arounds to eliminate or minimize the impact of these attacks.

[3.1](#) Blind connection-reset attacks

The Host Requirements RFC [[4](#)] states that a host SHOULD abort the corresponding connection when receiving an ICMP error message that indicates a hard error.

For ICMP messages of type 2 (Destination Unreachable) code 2 (protocol unreachable), specifically, the Host Requirements RFC states that even those transport protocols that have their own mechanisms to indicate that a port is unreachable MUST accept these ICMP error messages for the same purpose. That is, they MUST abort the corresponding connection when an ICMP port unreachable message is received.

Thus, an attacker could use ICMP to perform a blind connection-reset attack. That is, even being off-path, an attacker could reset any TCP connection taking place. In order to perform such an attack, an attacker would send any ICMP error message that indicates a "hard error", to either of the two TCP endpoints of the connection. Because of TCP's fault recovery policy, the connection would be immediately aborted.

All an attacker needs to know to perform such an attack is the socket pair that identifies the TCP connection to be attacked. In some scenarios, the IP addresses and port numbers in use may be easily guessed or known to the attacker [[6](#)].

There are some points to be considered about this type of attack:

o The source address of the ICMP error message need not be forged.
   Thus, simple egress-filtering based on the source address of IP
   packets would not serve as a counter-measure against this type of
   attack.

o Even if TCP itself were protected against the blind
   connection-reset attack described in [6] and [8], the type of
   attack described in this document could still succeed.

3.2 Degrading the performance of a connection

The Host requirements RFC states hosts MUST react to ICMP Source
Quench messages by slowing transmission on the connection. Thus, an
attacker could send ICMP Source Quench [2] messages to a TCP endpoint
to make it reduce the rate at which it sends data to the other party.
While this would not reset the connection, it would certainly degrade
the performance of the data transfer taking place over it.

4. Constraints in the possible solutions

The original ICMP specification [2] requires nodes generating ICMP
errors to include the IP header of the packet that elicited the ICMP
error message, plus the first 64 bits of its payload, in the payload
of the ICMP error message. For TCP, that means that the only fields
that will be included are: the source port number, the destination
port number, and the 32-bit sequence number. This imposes a
constraint on the possible solutions, as there is not much
information avalable on which to perform additional security checks.
While there exists a proposal to recommend hosts to include more data
from the original datagram in the payload of ICMP error messages
[13], we cannot yet propose any work-around based on any data past
the first 64 bits of the payload of the original IP datagram that
elicited the ICMP error message.

5. Solutions to the problem

There are a number of counter-measures against this type of attack.
Rather than being alternative measures, they could be implemented
together to increase the protection against this type of attack.

5.1 TCP sequence number checking

TCP SHOULD check that the sequence number in the TCP header contained
in the payload of the ICMP error message is within the range SND.UNA
< SEG.SEQ < SND.NXT. This means that the sequence number should be

within the range of the data already sent but not yet acknowledged.
If an ICMP error message doesn't pass this check, it SHOULD be
discarded.

Even if an attacker were able to guess the four-tuple that identifies
the TCP connection, this additional check would reduce the
possibility of success of the attacker to Flight_Size/2^^32 (where
Flight_Size is the number of data bytes already sent to the remote
peer, but not yet acknowledged [14]). For connections in the
SYN-SENT or SYN-RECEIVED states, this would reduce the probability of
success of a blind-connection reset attack during the
connection-establishment phase to 1/2^^32. For a TCP endpoint with
no data "in flight", this would completely eliminate the possibility
of success of this attack.

5.2 Delaying the connection-reset

For connections in any of the synchronized states, an additional
counter-measure against the blind connection-reset attack could be
taken. Rather than immediately aborting a connection, a TCP could
abort a connection only after an ICMP error message indicating a hard
error has been received a specified number of times, and the
corresponding data have already been retransmitted more than some
specified number of times.

For example, hosts could abort connections only after a fourth ICMP
error message indicating a hard error is received, and the
corresponding data have already been retransmitted more than six
times.

The rationale behind this proposed fix is that if a host can make
forward progress on a connection, it can completely disregard the
"hard errors" being indicated by the received ICMP error messages.

5.3 Changing the handling of ICMP error messages for connections in the synchronized states

An analysis of the circumstances in which ICMP messages that indicate
hard errors may be received can shed some light to minimize (or even
eliminate) the impact of blind connection-reset attacks.

5.3.1 ICMP type 2 (Destination Unreachable), code 2 (protocol unreachable)

This ICMP error message indicates that the host sending the ICMP
error message received a packet meant for a transport protocol it
does not support. For connection-oriented protocols such as TCP, one
could expect to receive such an error as the result of a connection

request. However, it would be strange to get such an error during
the life of a connection, as this would indicate that support for
that transport protocol has been removed from the host sending the
error message during the life of the corresponding connection. Thus,
it would be fair to ignore ICMP protocol unreachable error messages
meant for connections that are in synchronized states. For TCP, this
would mean one would ignore ICMP port unreachable error messages
meant for connections that are in the ESTABLISHED, FIN-WAIT-1,
FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK or TIME-WAIT states.

5.3.2 ICMP type 2 (Destination Unreachable), code 3 (port unreachable)

This error message indicates that the host sending the ICMP error
message received a packet meant for a socket (IP address, port
number) on which there is no process listening. Those transport
protocols which have their own mechanisms for notifying this
condition should not be receiving these error messages. However, the
Host Requirements RFC [4] states that even those transport protocols
that have their own mechanism for notifying the sender that a port is
unreachable MUST nevertheless accept an ICMP Port Unreachable for the
same purpose. For security reasons, it would be fair to ignore ICMP
port unreachable messages that are meant for protocols that have
their own mechanisms for reporting this condition.

5.3.3 ICMP type 2 (Destination Unreachable), code 4 (fragmentation needed and DF bit set)

This error message indicates that an intermediate node needed to
fragment a datagram, but the DF (Don't Fragment) bit in the IP header
was set. Those TCP/IP stacks implementing the Path-MTU Discovery
(PMTUD) mechanism [5] should not abort the corresponding connection
when such an error message is received.

5.4 Ignoring ICMP Source Quench messages

The Host Requirements RFC [4] states that hosts MUST react to ICMP
Source Quench messages by slowing transmission on the connection.
However, as discussed in the Requirements for IP Version 4 Routers
RFC [15], research seems to suggest ICMP Source Quench is an
ineffective (and unfair) antidote for congestion. Thus, we recommend
hosts to completely ignore ICMP Source Quench messages.

5.5 Port randomization

As discussed in the previous sections, in order to perform any of the
attacks described in this document, an attacker needs to guess (or
know) the four-tuple that identifies the connection to be attacked.
Randomizing the ephemeral ports used by the clients would reduce the

chances of success by an attacker.

A proposal exists to enable TCP to reassign a well-known port number to a random value [16].

5.6 Authentication

Hosts could require ICMP error messages to be authenticated [12], in order to act upon them. However, while this requirement could make sense for those ICMP error messages sent by hosts, it would not be feasible for those ICMP error messages generated by intermediate routers.

[12] contains a discussion on the authentication of ICMP messages.

6. Future work

The same considerations discussed in this document should be applied to other similar protocols.

7. Security Considerations

This document describes the use of ICMP error messages to perform a number of attacks against the TCP protocol, and proposes a number of counter-measures that either eliminate or reduce the impact of these attacks.

8. Acknowledgements

This document was inspired by Mikka Liljeberg, while discussing some issues related to [17] by private e-mail. The author would like to thank James Carlson, Juan Fraschini, Markus Friedl, Guillermo Gont, Michael Kerrisk, Kacheong Poon, and Andrew Powell, for contributing many valuable comments.

9. References

9.1 Normative References

[1] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[2] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[4]   Braden, R., "Requirements for Internet Hosts - Communication
      Layers", STD 3, RFC 1122, October 1989.

[5]   Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191,
      November 1990.

9.2 Informative References

[6]   Watson, P., "Slipping in the Window: TCP Reset Attacks", 2004
       CanSecWest Conference , 2004.

[7]   Jacobson, V., Braden, B. and D. Borman, "TCP Extensions for
       High Performance", RFC 1323, May 1992.

[8]   Stewart, R., "Transmission Control Protocol security
       considerations", draft-ietf-tcpm-tcpsecure-01 (work in
       progress), June 2004.

[9]   Touch, J., "ANONsec: Anonymous IPsec to Defend Against Spoofing
       Attacks", draft-touch-anonsec-00 (work in progress), May 2004.

[10]   Poon, K., "Use of TCP timestamp option to defend against blind
       spoofing attack", draft-poon-tcp-tstamp-mod-00 (work in
       progress), June 2004.

[11]   Clark, D., "Fault isolation and recovery", RFC 816, July 1982.

[12]   Kent, S. and R. Atkinson, "Security Architecture for the
       Internet Protocol", RFC 2401, November 1998.

[13]   Gont, F., "Increasing the payload of ICMP error messages",
       (work in progress) draft-gont-icmp-payload-00.txt, 2004.

[14]   Allman, M., Paxson, V. and W. Stevens, "TCP Congestion
       Control", RFC 2581, April 1999.

[15]   Baker, F., "Requirements for IP Version 4 Routers", RFC 1812,
       June 1995.

[16]   Shepard, T., "Reassign Port Number option for TCP",
       draft-shepard-tcp-reassign-port-number-00 (work in progress),
       July 2004.

[17]   Gont, F., "TCP's Reaction to Soft Errors",
       draft-gont-tcpm-tcp-soft-errors-00 (work in progress), June
       2004.

Author's Address

    Fernando Gont
    Universidad Tecnologica Nacional
    Evaristo Carriego 2644
    Haedo, Provincia de Buenos Aires 1706
    Argentina

    Phone: +54 11 4650 8472
    EMail: fernando@gont.com.ar

Appendix A. **Changes from draft-gont-tcpm-icmp-attacks-00**

    o Added Section 5.3

    o Added a summary of the relevant RFCs in several sections

    o Miscellaneous editorial changes