Network Working Group
Internet-Draft
Expires: January 8, 2005

M. Mathis
J. Heffner
B. Chandler
PSC
July 10, 2004

# Fragmentation Considered Very Harmful
# draft-mathis-frag-harmful-00

## Status of this Memo



## Copyright Notice



## Abstract

IPv4 fragmentation is not sufficiently robust for general use in today's Internet. The 16-bit IP identification field is not large enough to prevent frequent missassociated IP fragments and the TCP and UDP checksums are insufficient to prevent the resulting corrupted data from being delivered to higher protocol layers. In this note we describe some easily reproduced experiments demonstrating the problem and estimate the scale the data corruption in the presence of ever growing data rates.

## 1. Introduction

The IPv4 header was designed at a time when data rates were several orders of magnitude lower than those achievable today. In this document, we describe a consequent scale-related failure in the IP identification (ID) field, where fragments may be mis-associated at a rate high enough likely to invalidate assumptions about data integrity failure rates. We also outline scenarios in which data corruption may happen reliably and reproducibly.

While a number of problems with IP fragmentation have been well documented [1], this presents a relatively new and serious operational problem given the severity of the failure mode, and that it occurs on what is today common communications equipment. It is especially pertinent due to the recent proliferation of UDP bulk transport tools which do not do MTU discovery , and some network equipment which ignores the Don't Fragment (DF) bit in the IP header as a work-around for MTU discovery problems [2].

## 2. Wrapping the IP ID Field

The Internet Protocol standard specifies:

> "The choice of the Identifier for a datagram is based on the need to provide a way to uniquely identify the fragments of a particular datagram. The protocol module assembling fragments judges fragments to belong to the same datagram if they have the same source, destination, protocol, and Identifier. Thus, the sender must choose the Identifier to be unique for this source, destination pair and protocol for the time the datagram (or any fragment of it) could be alive in the internet." [3]

Strict conformance to this standard limits transmissions in one direction between any address pair to no more than 65536 datagrams per maximum packet lifetime.

Obviously hosts do not follow the standard so strictly. Assuming a maximum packet lifetime on the order of seconds, today it is common for host interfaces to send at rates higher than this. For example, a host with a 100 Mbps interface sending 1500 byte packets may send 65536 packets in under 8 seconds.

The problem occurs when a fragment is dropped by the network, and a later fragment is received that, while part of a different datagram, has the same ID value and fragment offset as the dropped fragment. The two fragments will be incorrectly spliced together and delivered to the layer above IP. It is common that the fragment offset and length would match since packets of the same size sent along the same

path will be fragmented in the same manner. In 65537 segments, there must be at least two with matching ID fields. If the sender is transmitting segments fast enough that datagrams are send with duplicate ID fields within the reassembly timeout (a suggested value is 15 seconds [3]), then fragments may be mis-associated.

The case of particular concern occurs when only the first fragment of a datagram is lost by the network. The remaining fragments will be stored in the fragment reassembly buffer, and at some point in the future a new packet will arrive with the matching ID field. This new first fragment will be (incorrectly) matched up with the rest of the old packet and delivered to the upper layer. Assuming the fragments are delivered in order, the rest of the new datagram will be buffered, forming a cycle. One of every 65536 datagrams will be incorrectly reassembled by the IP layer. It is possible to have a number of simultaneous cycles, bounded by the size of the fragment reassembly buffer.

Most TCP implementations today participate in MTU discovery [4], which will avoid this problem by avoiding fragmentation. However, as a work-around for MTU discovery problems [2], some TCP implementations and communications gear provide mechanisms to disable path MTU discovery by clearing or ignoring the DF bit.

## 3. Harmful Effects of Mis-associated Fragments

When the mis-associated fragments are delivered, transport-layer checksumming should detect these datagrams as incorrect and discard them. When the datagrams are discarded, it could pose a problem for loss feedback congestion control algorithms since there will be a high number of non-congestion-related losses.

However, transport checksums may not be designed to handle such high error rates, either. The UDP checksum is only 16 bits in length. If these checksums follow a uniform random distribution, we expect mis-associated datagrams to be accepted by the checksum at a rate of one per 65536. With only one mis-association cycle, we expect corrupt data delivered to the application layer once per 2^32 datagrams. This number can be significantly higher with multiple cycles.

With non-random data, the UDP checksum may be even weaker still. It is possible to construct datasets where mis-associated fragments will always have the same checksum. Such a case may be considered unlikely, but is worth considering. "Real" data may be more likely than random data to cause checksum hotspots and increase the probability of false checksum match [5]. Also, some applications may turn off checksumming to increase speed, though this practice has

been found to be dangerous for other reasons [6].

## 4. Experimental Results

To test the practical impact of fragmentation on UDP, we ran a series of experiments with a common UDP bulk transport protocol, Reliable Blast UDP (RBUDP), part of the QUANTA networking toolkit. It is one of the tools used as an alternative to TCP for high-bandwidth applications on specialized networks. The choice to use RBUDP has very little to do with the protocol itself, as any UDP transport tool without extra corruption detection would work equally well.

In order to diagnose corruption on files transferred with RBUDP, we used a file format including embedded sequence numbers and MD5 checksums. These were placed such that one set was included in each fragment of each datagram. Thus it was possible to distinguish random corruption from that caused by mis-associated fragments.

Two types of dataset were used. In the first, all space not used for sequence numbers and MD5 checksums was filled with pseudo-random data, giving datagrams random checksums. The second was constructed in a similar manner except that the upper halves of each 32-bit word were filled with the 16-bit ones complement of the lower half. This gave each 32-bit word a zero ones-complement sum, so datagrams had constant checksums. With these constant checksums, mis-associated fragments were guaranteed not to fail the UDP checksum test. Each dataset used was 400 MB in size.

The RBUDP tools were used to send the datasets between a pair of hosts at slightly less than the available datarate. Near the beginning of each flow, a brief secondary flow was started to induce packet loss in the primary flow. Throughout the life of the primary flow, we typically observed mis-association rates on the order of 0.05%. In datasets with constant checksums, each of these mis-associations resulted in corrupted data. In sending datasets with random checksums 100 times (for a total of 100 GB), we observed one corruption and 41091 bad UDP checksums.

## 5. Remedies

IPv6 is less vulnerable to this type of problem, since its fragment header contains a 32-bit identification field [7]. Mis-association will only be a problem at packet rates 65536 times higher than for IPv4.

Since mis-association of fragments will only occur when the IP ID field is wrapped within the fragment reassembly timeout, it is possible to reduce the timeout so that this situation is less likely

to occur. Since the timeout is set by the receiving host while the IP ID field is set by the sending host, it is not generally possible to set the timeout low enough so that a fast sender's fragments will not be mis-association, yet high enough so that a slow sender's fragments will not be unconditionally discarded before it is possible to reassemble them. It is not within the scope of this document to recommend timeout values.

Another means of solving the corruption issue is to add stronger integrity checking, which can be done at any layer above IP. This is a natural side effect of using cryptographic authentication. At the network layer, if IPsec AH is in use, the mis-associated fragments should be discarded with extremely high probability. Other higher layers may use longer checksums (for example, SCTP's is 32 bits in length [8]) or cryptographic authentication (SSH message authentication codes [10]). While stronger integrity checking may prevent data corruption, it will not solve the problem of a high effective loss rate.

## 6. Security Considerations

If a malicious entity knows that a pair of hosts are communicating using a fragmented stream, it may present an opportunity for this entity to corrupt the flow. By sending "high" fragments (those with offset greater than zero) with a forged source address, the attacker can deliberately cause corruption as described above. Exploiting this vulnerability requires only knowledge of the source and destination addresses of the flow, and fragment boundaries. It does not require knowledge of port or sequence numbers.

If the attacker has visibility of packets on the path, the attack profile is similar to injecting full segments. Using this attack makes blind disruptions easier, and could certainly be used effectively to cause denial of service. However, only streams using IPv4 fragmentation are vulnerable. Because of the nature of the problems outlined in this draft, the use of IPv4 fragmentation for critical applications may not be advisable regardless of security concerns.

## 7 References

[1] Kent, C. and J. Mogul, "Fragmentation considered harmful", Proc. SIGCOMM '87 vol. 17, No. 5, October 1987.

[2] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, September 2000.

[3] Postel, J., "Internet Protocol", STD 5, RFC 791, September

1981.

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

[5] Stone, J., Greenwald, M., Partridge, C. and J. Hughes, "Performance of Checksums and CRC's over Real Data", IEEE/ACM Transactions on Networking vol. 6, No. 5, October 1998.

[6] Stone, J. and C. Partridge, "When The CRC and TCP Checksum Disagree", Proc. SIGCOMM 2000 vol. 30, No. 4, October 2000.

[7] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[8] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[9] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[10] Ylonen, T. and C. Lonvick, "SSH Transport Layer Protocol", draft-ietf-secsh-transport-18 (work in progress), June 2004.

[11] Clark, D., "IP datagram reassembly algorithms", RFC 815, July 1982.

## Authors' Addresses

Matt Mathis
Pittsburgh Supercomputing Center
4400 Fifth Avenue
Pittsburgh, PA  15213
US

Phone: 412-268-3319
EMail: mathis@psc.edu

John W. Heffner
Pittsburgh Supercomputing Center
4400 Fifth Avenue
Pittsburgh, PA  15213
US

Phone: 412-268-2329
EMail: jheffner@psc.edu

Ben Chandler
Pittsburgh Supercomputing Center
4400 Fifth Avenue
Pittsburgh, PA  15213
US

Phone: 412-268-9783
EMail: bchandle@psc.edu

## Appendix A. Support

This work was supported by the National Science Foundation under
Grant No. 0083285.

## Intellectual Property Statement



## Disclaimer of Validity



## Copyright Statement



## Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.