IPV6 FRAGMENTATION

The Case For Deprecation

Ron Bonica

NANOG58

BACKGROUND

STATUS QUO

In order to send a packet larger than the PMTU, an IPv6 node may fragment a packet at the source and have it reassembled at the destination

- In IPv6, only hosts can fragment
- In IPv4, both hosts and routers can fragment

IPv6 Fragmentation has always been discouraged

- Reassembly is computationally expensive and inefficient
- Security concerns

SECURITY CONCERNS

DoS attacks

- Attacker sends fragmented packets to victim
  - Attack flow is optimized to consume resources on victim platform
- Attacker spoofs PTB message to victim's legitimate communication partners
  - Causes legitimate communication partners to fragment packets that don't need to be fragmented

Evasion of stateless firewall filters

- Stateless firewall selects packets based upon fields drawn from both the IP and TCP headers
- Attacker fragments packets so that IP header is in first fragment and TCP header is in second fragment
- All fragments evade selection by firewall
- draft-ietf-6man-oversized-header-chain

EXPOSING BUGS IN RARELY EXERCISED BRANCHES OF REASSEMBLY CODE

Implementations occasionally deal badly with the following

- Fragment overlap
- Fragment overwrite
- Fragment overrun
- Too many fragments being reassembled simultaneously
- Too many packets that cannot be reassembled due to missing fragments

The best implementations deal with these effectively

But sometimes they don’t

- Rarely exercised code on the OS should concern everyone

A (BAD) ALTERNATIVE TO IPV6 FRAGMENTATION

All upper layers send packets smaller that 1280 bytes all of the time

Works in the vast majority of cases

- Exception: In response to an IPv6 packet that is sent to an IPv4 destination, the originating IPv6 node may receive an ICMP Packet Too Big message reporting a Next-Hop MTU less than 1280

Hammer is way too big

A BETTER ALTERNATIVE IPV6 FRAGMENTATION

An upper layer executes PMTUD [RFC 1981] or PLMTUD [RFC 4821] procedures

- Moves problems of fragmentation and reassembly from the IP layer to an upper layer
  - There is no free lunch!

Many TCP implementations support PMTUD and/or PLMTUD

According to RFC 5405, a UDP-based application SHOULD NOT send UDP datagrams that result in IP packets exceeding the PMTU. The application should do one of the following:

- Use the path MTU information provided by the IP layer
- Implement PMTUD/PLMTUD itself
- Send only packets known not to exceed the PMTUD

THE BENEFIT OF PMTU/PLMTUD DISCOVERY

Moves the problems of fragmentation and reassembly from the IP layer to an upper layer

- Either the transport or application layer
- Called a “packetization layer”

Localizes risk

Allows for layer specific optimizations

- Example: A particular packetization layer knows that it will never send a packet longer than 1280 bytes

OPERATIONAL REALITY

FRAGMENTED IPV6 TRAFFIC IS RARE

Most popular TCP implementation perform PMTUD or PLMTUD procedures

- So, applications that ride over TCP rarely cause fragments to be sent

Many UDP-based applications abide by the recommendations of RFC 5405

A few important UDP-based applications do not abide by the recommendations of RFC 5405

- Example: DNSSEC can send large UDP packets. TCP alternative available

THE BITTER TRUTH

Many operators discard fragmented IPv6 packets

An NLnet Labs Study* reveals that

- IPv4 fragments were discarded along ~ 12% of observed paths
- IPv6 fragments were discarded along ~ 40% of observed paths

So, if you are sending IPv4 and/or IPv6 fragments, they may not make it to their destination!

* http://www.nlnetlabs.nl/downloads/publications/pmtu-black-holes-msc-thesis.pdf

RECOMMENDATION

A STANDARDS TRACK RFC (UPDATES RFC 2460)

Deprecates the IPv6 Fragment Header

- Please, don’t write any new applications that fragment packets
- Existing applications will continue to work
  - As well or poorly as the do today

States that operators MAY discard packets containing the IPv6 Fragment Header

- As, in fact, they already do

everywhere