Deep Dive into IPv6 Extension Header Testing on Cloud Platforms

draft-elkins-v6ops-eh-deepdive-cloud-00

Nalini Elkins: Inside Products, Inc.
Dr. Priyanka Sinha: Independent
Arneya Deshpande: Google
Vivek Jain: UC Riverside
Our Goals

- We are doing troubleshooting and trying to isolate and fix problems.
- We are NOT doing large scale measurements.
- We will do that after problems are fixed. Otherwise, moving target.
- Two large providers of Internet services appear to have changed their EH handling already. Conversations with a two cloud vendors in progress. They stand ready to work with us.
Our Testing Platform

• Used a small hosting service (not one of the “brand-name” ones).

• All machines are Ubuntu using eBPF. An IPv6 Destination Header (PDM) was sent with every packet.

• Using Apache Web Server. So, real traffic sent to and from cloud.
Interim Definition of Cloud

- Service / infrastructure provided by vendors such as Google, Microsoft, Oracle, AWS, et al.

- Let’s work this out later.
Cloud Topologies

Various configurations:
- Outside Cloud to Inside Cloud
  - Standalone to Cloud (OC-S)
  - Cloud to Standalone (S-OC)
  - Data center to Cloud (OC-D)
- Inside Cloud (IC),
  - Cloud #1 to One Datacenter (IC-SD)
  - Cloud #1 to Multiple Datacenters (IC-MD)
- Between Clouds (BC)
  - Cloud#1 to Cloud #2
Bottom Line

- IPv6 to Cloud Provider #1 with no IPv6 extension headers works fine. (PING and HTTP). Client can be inside cloud or outside cloud.
- IPv6 to Cloud Provider #1 with IPv6 extension headers does not work. Client can be inside cloud or outside cloud.
  - OS supports EHs
  - “Network” does not support EHs
- Inside Cloud: IPv6 in Cloud Provider #1 with IPv6 extension headers works
  - OS supports
  - Internal “Network” appears to be not a factor
  - Link Local as well as Global Unicast works
"Network" does not WHAT?

Bill Jouris, 2023-07-20T17:56:23.981
Standalone to Cloud (OC-S) Topology
Simplest: Client -- Internet -- Server
Realistic Topology
Client to Internet to Cloud Network

This may be the problem!
Cloud Provider #2

- Seems to work the same way but with one exception!

- ICMPv6 Checksum not computed correctly. (We believe!)  

- They use an “external” IPv6 address and an “internal” IPv6 address
What happened?

• The initial checksum is correctly computed at Cloud Provider #2.

• However, ICMPv6 uses a 16-bit pseudo-header checksum field -- IPv6 source and destination addresses, etc.

• The “network” changes the source address to 2603:1030:20e:3::369 but does not rebuild the checksum.

So, here’s the potential bug.
- The load balancer finds a Next Header field that is not ICMP, TCP, or UDP and it doesn’t “follow” the NH chain to find if there is a L4 protocol with a checksum that needs to be updated.
Next Steps …

• Get bugs fixed!
• Test with:
  • More cloud providers
  • Routers
  • ISPs
  • Load balancers
  • OSs
• Need to test ALL extension headers!
• This will be a multi-year process!
• Happy to talk to anyone offline to review traces!
Supporting Slides
Cloud Testing: Outside Cloud to Inside Cloud
Cloud #1 to Standalone outside Cloud (S-OC)
Cloud Client to Internet to External Server

Cloud
Client

Internet

2600:1900:41a0:71b6:0:1::

Server
2001:1900:5:3ce7:5400:4ff:fe31:1527
Packet trace with No Extension Headers

Cloud is client. Going to Standalone server outside Cloud. No EH.
WHO-IS LookUp shows that this is cloud provider #1.

<table>
<thead>
<tr>
<th>Network</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Net Range</strong></td>
<td><strong>2600:1900:: - 2600:1900/28</strong></td>
</tr>
<tr>
<td><strong>CIDR</strong></td>
<td><strong>2600:1900::/28</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td><strong>CLOUD</strong></td>
</tr>
<tr>
<td><strong>Handle</strong></td>
<td><strong>NET6-2600-1900-1</strong></td>
</tr>
<tr>
<td><strong>Parent</strong></td>
<td><strong>NET6-2600 (NET6-2600-1)</strong></td>
</tr>
<tr>
<td><strong>Net Type</strong></td>
<td><strong>Direct Allocation</strong></td>
</tr>
<tr>
<td><strong>Origin AS</strong></td>
<td><strong>AS</strong></td>
</tr>
<tr>
<td><strong>Organization</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Registration Date</strong></td>
<td><strong>2014-06-28</strong></td>
</tr>
<tr>
<td><strong>Last Updated</strong></td>
<td><strong>2015-09-21</strong></td>
</tr>
<tr>
<td><strong>Comments</strong></td>
<td><strong>&quot;The IP addresses under this netblock are in use by Cloud customers&quot;</strong></td>
</tr>
</tbody>
</table>

Direct all copyright and legal complaints to https://support.client.com/legal/ip/report
What do we know so far?

- IP addressing and firewalls set up correctly
- HTTP server set up correctly
- Traffic going back and forth
Bottom Line

- IPv6 to Cloud Provider #1 with no IPv6 extension headers works fine. (PING and HTTP). Client can be inside cloud or outside cloud.
Let's add EHs

Cloud is client (:1::). Going to Standalone server outside Cloud. Client has EH (:1527).
What is received at other end?

- Nothing!
- (Only IPv4 packets!)
Cloud is Server (:1::).

Client is Standalone outside Cloud. Client has EH (:1527).

Let's try the other way.
What is received at other end?

- Nothing!
- (Only IPv4 packets!)
Bottom Line

- IPv6 to Cloud Provider #1 with no IPv6 extension headers works fine. (PING and HTTP). Client can be inside cloud or outside cloud.
- IPv6 with Cloud Provider #1 with IPv6 extension headers does not work. Client can be inside cloud or outside cloud.
  - OS supports EHs
  - “Network” does not support EHs
Realistic Topology
Client to Internet to Cloud Network

Client — Internet

This may be the problem!
Cloud Testing: Inside Cloud
Inside Cloud (IC)
Cloud #1 to One Datacenter (IC-SD) Client to Server
WHO-IS LookUp shows that this is cloud provider #1.

<table>
<thead>
<tr>
<th>Network</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIDR</td>
<td>2600:1900::/28</td>
</tr>
<tr>
<td>Name</td>
<td>CLOUD</td>
</tr>
<tr>
<td>Handle</td>
<td>NET6-2600-1900-1</td>
</tr>
<tr>
<td>Parent</td>
<td>NET6-2600 (NET6-2600-1)</td>
</tr>
<tr>
<td>Net Type</td>
<td>Direct Allocation</td>
</tr>
<tr>
<td>Origin AS</td>
<td>AS</td>
</tr>
<tr>
<td>Organization</td>
<td></td>
</tr>
<tr>
<td>Registration Date</td>
<td>2014-05-28</td>
</tr>
<tr>
<td>Last Updated</td>
<td>2015-09-21</td>
</tr>
<tr>
<td>Comments</td>
<td><strong>The IP addresses under this netblock are in use by Cloud customers</strong></td>
</tr>
</tbody>
</table>

Direct all copies of illegal complaints to https://support.cloudflare.com/hc/en-us/report
WHO-IS LookUp shows that this is cloud provider #1.
Inside Cloud (IC)
Nominal: Cloud #1 to One Datacenter (IC SD)
Realistic: Client to Load Balancer, Firewall, etc. to Server
Ping from Inside Cloud with EH works fine.
As does HTTP.
Let's look at Link Local in Cloud

Unicast Link Local. One LL has EH, other does not. Works fine. Look at the XID and CID.
Response comes back fine.

From the other side Link Local
Bottom Line

- IPv6 to Cloud Provider #1 with no IPv6 extension headers works fine. (PING and HTTP). Client can be inside cloud or outside cloud.
- IPv6 to Cloud Provider #1 with IPv6 extension headers does not work. Client can be inside cloud or outside cloud.
  - OS supports EHs
  - “Network” does not support EHs
- Inside Cloud: IPv6 in Cloud Provider #1 with IPv6 extension headers works
  - OS supports
  - Internal “Network” appears to be not a factor
  - Link Local as well as Global Unicast works
Cloud #1 – Multiple Datacenters (IC-MD)

Multi Data Center
Cloud may not work the same way
Cloud Provider #2

- Seems to work the same way but with one exception!
- ICMPv6 Checksum not computed correctly.
- They use an “external” IPv6 address and an “internal” IPv6 address
Packet trace with Extension Headers

Capture from Cloud provider 2

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
</table>

**Next Header:** Destination Options for IPv6 (58)

- Hop Limit: 66
- Source Address: 2003:1099:100:1:100

**Destination Options for IPv6**

- **Type:** Echo (ping) request (128)
- **Code:** 0
- **Checksum:** 0x60 (Incorrect, should be 0x85a2)
- **Identifier:** 0x0001
- **Sequence:** 1
- **Required:** 0
- **Options:** 0
- **Data (56 bytes):**

**Extension Headers**

- **External IPv6 address**
- **Checksum is using “internal” IPv6 address**
What happened?

• The initial checksum is correctly computed at Cloud Provider #2.

• However, ICMPv6 uses a 16-bit pseudo-header checksum field -- IPv6 source and destination addresses, etc.

• The “network” changes the source address to 2603:1030:20e:3::369 but does not rebuild the checksum.

So, here’s the potential bug.
• The load balancer finds a Next Header field that is not ICMP, TCP, or UDP and it doesn’t “follow” the NH chain to find if there is a L4 protocol with a checksum that needs to be updated.
Any transport or other upper-layer protocol that includes the addresses from the IP header in its checksum computation must be modified for use over IPv6, to include the 128-bit IPv6 addresses instead of 32-bit IPv4 addresses.
Next Steps ...

- Get bugs fixed!
- Test with:
  - More cloud providers
  - Routers
  - ISPs
  - Load balancers
  - OSs
- Need to test ALL extension headers!
- This will be a multi-year process!
- Happy to talk to anyone offline to review traces!
RFP for Infrastructure Services (FYI)

• The IETF Administration LLC is soliciting bids for Infrastructure Services.

• The current contract for IETF IT infrastructure services is a black box contract - we specify the systems to be maintained along with a very basic SLA, and the provider is responsible for the underlying infrastructure on which those systems operate, including the system administration strategy. This underlying infrastructure consists of a small number of managed servers with most applications installed directly onto those servers though more recently containers have been used.

• The IETF Administration LLC has consulted with the community to develop a new operational strategy for how the infrastructure should be operated. This strategy sets goals for the infrastructure to move to the cloud and to be managed very differently. As well as providing for a more modern infrastructure, this new strategy also lays the foundations for a change to the architecture of our in-house applications to take advantage of modern scaling and hosting capabilities.

• This RFP is for a service provider to design the new cloud based infrastructure, migrate the existing services to that infrastructure and then manage the infrastructure. It is likely that this management will involve occasional projects to support major changes in application deployment.