Semantic differences of IPv6

Francesco Prelz
INFN, sezione di Milano
(and members of the HEPix IPv6 group)

Summary

1. There are *semantic* differences between IPv4 and IPv6.
    - That prevent applications from being *transport-layer agnostic*
    - even once all *syntactic* differences are taken into account (e.g. different address length, looking for a port number after the first colon in an address string, parsing 'defa(ult)' as a hex number, etc.)
2. Some of these have implications on security.
3. Knowledge of these differences allows to gauge the complexity of certain IPv6 porting issues outlined in subsequent talks.

Main semantic differences of IPv6 (1)

1. Every network end-point is *always* associated to *multiple* active network addresses.

- At least the *link-local* address on top of any larger or global scope address.
- ⇒ An origin *and* a destination address has to be *chosen* for every network communication.
- Preferences for this choice are *new* system configuration parameters (Linux: `/etc/gai.conf` and `ip -6 addrlabel list`).
- All of this is documented in RFC6724: not easy reading material by any standard.
- The notion of a 'main' network address for a network interface disappears (except for Mobile IP, where the *Home Address* is always preferred).
- Whether IPv4 o IPv6 should be preferred on a dual-stack system is just a special case of this choice.
    - Painlessly keeping the preference for IPv6 everywhere is essential to prevent the transition train from derailing. Moving to RHEL6 and derivatives has already shown one instance (listening sockets with `PF_UNSPEC`) where the preference for IPv6 was lost, and dual-stack operation followed suit.
    - This derived from the conflicting provisions of two RFCs: 4291 and 6724. The only solution short of relinquishing hybrid stacks (or correcting many applications) would be an amendment of RFC6724.

Main semantic differences of IPv6 (2)

2. IP *route* assignment.

- IPv6 route assignment can only occur either statically or via *Router Advertisements*, and these are *multicast* messages.
  - This is the other way around w.r.t. DHCP(v4): *push* vs. *pull*.
- Router Advertisements should originate from *one* legitimate network source (unless one wants to experience the scenarios described in RFC6104) and instruct *all* hosts listening on a network segment:
  - To enable (or disable) autoconfiguration of their IP addresses (autonomous bit in prefix announcements).
  - To optionally seek other configuration information via a DHCPv6 query (managed and other bits in Router Advertisements).
  - To add to their routing tables an arbitrary number of IP routes (including the default route), with a given lifetime.
- The default route *cannot* be assigned via DHCPv6.
  Five IETF drafts (1, 2, 3, 4, 5) proposing to add this option expired.
  This seems to be a hopeless proposition.

Main semantic differences of IPv6 (3)

3. IP address and DNS server assignment.

- There are mechanisms (beyond manual assignment, which is always an option) for simple, automatic and painless (?) configuration of the transport layer:
  1. *Stateless* autoconfiguration or SLAAC (basically: hosts append their MAC address to a prefix that is received by the local router via multicast). However, RFC6106, that would allow to configure the DNS servers via Router Advertisements as well, doesn't seem to be widely supported.
  2. *Stateful* autoconfiguration or DHCPv6 (clients are identified by two numbers that are generated *at run-time*: the DUID identifies the host while the IAID identifies each interface). As already remarked, distribution of Router Advertisements (and protection against *rogue RAs*) is still needed.
- One has to take care at least of:
  1. Updating the DNS with the reverse resolutions of assigned addresses.
  2. Keeping the ability to associate network traffic to a legitimate and properly authorised user (*last-hop traceback*), as required by various policies and laws.

IPv6 differences with an impact on security (1)

- RFC 4942 identifies three classes of issues with an impact of security (and then deals with the first one, as it is addressed to stack developers):
  1. Issues due to IPv6 protocol
  2. Issues due to Transition Mechanisms
  3. Issues due to IPv6 Deployment

Implementations are *required* by the protocol specs to process extension headers for functions that used to be optional under IPv4 (IPsec, QOS, mobile IP support). Other functions are re-implemented anew (DHCP, neighbor discovery, anycast, multicast).

- New code in network stacks is a wonderful chance for anyone looking for exploitable vulnerabilities. Only time and adoption will tell how many are there.

The address space is notoriously larger (128 bit - 340 undecilion addresses, 2^{120} possible unicast addresses, where 2^{112} public addresses in class `2001::` can be allocated already).

- A brute force scan is impractical. However, while it may be wise to allocate non-contiguous addresses, there are plenty of other methods to find possible attack targets in a given network.
- Address collision becomes a much more unlikely indicator of unauthorised address.

IPv6 differences with an impact on security (2)

Much functionality needed by the protocol (Neighbor Discovery, MTU discovery, Mobility) is now part of ICMPv6 at level 3. Level 3 communication *must* therefore automatically work at least for hosts on the same network link.

- There are *many* new messages in ICMPv6, and there is a RFC (4890) just to specify minimal firewall rules for IPv6 accesses.
- Neighbor Discovery operates on local multicast.
- ICMPv6 on the other hand explicitely prohibits to respond to/amplify requests that are sent to multicast addresses (e.g: `FF02::1`, all-link-local-nodes, `FF05::2`, all-routers, see RFC2373).

Intermediate equipment *is not allowed* to fragment packets.
The minimal MTU is 1280 bytes.

- Stack implementations could still harm themselves by creating smaller intermediate fragments.

IPv6 differences with an impact on security (3)

The large number of optional protocol headers may mean that upper-layer payload (with port numbers, and other information of potential use to filters and firewalls) *is not found in the first fragment* so that packet reassembly is required.

To conclude, and partially justify this (worrisome?) list of differences, it may be worth reminding that the IPv6 design was indeed pulled in all directions. Mainly towards:

1. restoring the internal transparency of the transport layer (by removing intermediate *stateful* elements such as NATs, Firewalls, etc.);
2. enabling an easier, and possibly automatic, configuration of the transport layer;
3. allowing IP mobile applications.

References

- HEPix IPv6 group knowledge base: `http://hepix-ipv6.web.cern.ch/knowledge-base`.
- Scott Hogg, Eric Vyncke, "IPv6 Security", Cisco Press, January 2009.
- Microsoft, "IPv6 Security Considerations and Recommendations ", last update August 2011
- Countless RFCs: 2373, 2375, 2460, 2461, 2463, 2766, 3041, 3315, 3484, 3971, 4191, 4291, 4860, 4861, 4862, 4890, 4942, 4966, 6104, 6106, 6724.