Making an application fully IPv6 compliant

A modest check list

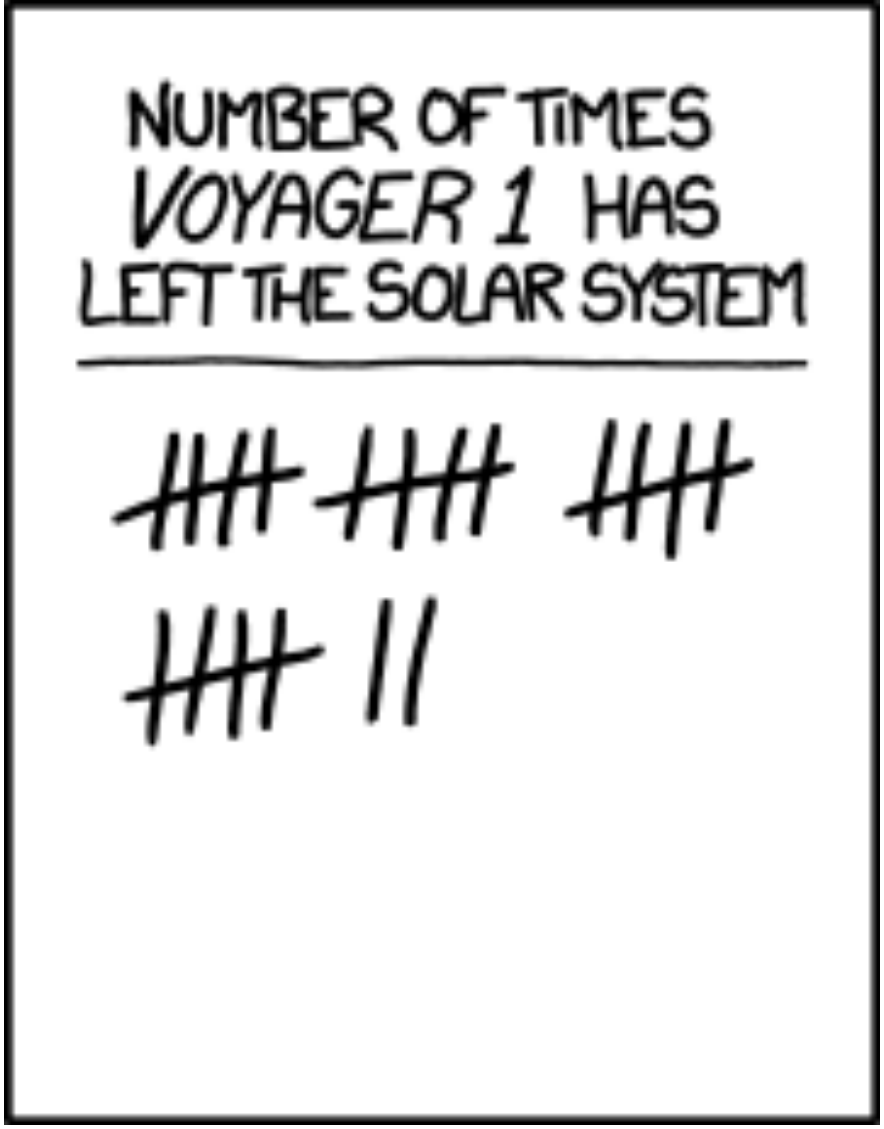
http://tinyurl.com/ipv6-checklist

bert.hubert@powerdns.com
+31-622-440-095

POWERDNS

Whoami & Goal

- Founder of PowerDNS, powers around 40% of all domain names in Western Europe, around 10% of resolvers, around 90% of DNSSEC
  - Open source, fully supported
- "Been fully IPv6 compliant since 2002"
- Modest goal of this presentation: plant seed in people's minds that *full* IPv6 support is more than adding AF_INET6
- Expected to be familiar with IPv4 ;-)
- No fireworks, but perhaps we can prevent..



So far Voyager 1 has "left the Solar System" by passing through the termination shock three times, the heliopause twice, and once each through the heliosheath, heliosphere, heliodrome, auroral discontinuity, Heaviside layer, trans-Neptunian panic zone, magnetogap, US Census Bureau Solar System statistical boundary, Kuiper gauntlet, Oort void, and crystal sphere holding the fixed stars

Where does IPv6 sit?

- Among the 'geekerati' of the world there is widespread disbelief that IPv6 isn't deployed more widely
- Partially, we can't control this, but partially, we might forget that "I listen on ::" is only part of the story
- IPv6 is also in: monitoring, firewalling, load balancing, IDS/IPS, virus scanners (!)
- Deployment only "easy" once it is in all bits, including the boring ones

Areas to look at

- Creating & binding to the socket!
- Converting host names & "presentation form addresses" into IPv4 and/or IPv6 addresses
  - Decide on presentation format for port, ie [::]:25
  - Pick policy for what it *means* if a host has 3 IPv4 and 1 IPv6 address
- Properly filling out sockaddr_in6, including 'scope'
  - And how/if to parse scope field in presentation format
- Security / resource exhaustion
- Filters, ACLs, proxies, web servers..

The socket

- Create a socket just like IPv4, now use AF_INET6
- Of note:
  - Some operating systems (FreeBSD, OpenBSD) can be compiled utterly without IPv6. Sometimes mandatory because of security guidelines. AF_INET6 might not even be defined or, you can't even resolve :: (even if you don't make a socket!)
  - Be able to deal with the inability to make an AF_INET6 socket!
- It IS possible to bind to IPv6 and get IPv4 connections on your socket too

IPv4 & IPv6 collisions

- The IPv4 'ANY' address 0.0.0.0 has a family member in [::] for IPv6
- Should you want to, you can setsockopt IPV6_V6ONLY to 0, and [::] will function like 0.0.0.0 too
  - Including IPv4 addresses mapped to IPv6 addresses, [::ffff:46.4.95.140]
- On Linux, IPV6_V6ONLY defaults to 0 so you CAN'T bind to 0.0.0.0 and [::] simultaneously by default
  - This is sad, but we have to deal with it -> just set it to 1 always

sockaddr, sockaddr_in, sockaddr_in6

- C's attempt at inheritance. Painful.
- It is guaranteed that sin_family field of sockaddr_in and sockaddr_in6 overlap in memory, so you can 'peek what you got'
- A union on of a sockaddr_in and a sockaddr_in6 works VERY WELL to pass around 'an IPv4 or IPv6 address' (link to blogpost at the end)
- **HOWEVER, need to provide length of sockaddr with sockets API & some operating systems will not accept longer IPv6 length for IPv4 socket**

sockaddr_in{,6} differences

- Contents of sockaddr_in are all well known (family, address, port number, that's it)
  - No need to zero it
- sockaddr_in6 has extra fields:
  - flowinfo
  - scope_id
  - ... who knows
- If you neglect to zero a sockaddr_in6, it will work **"most of the time"**
- Does anybody know what a flowinfo is?
  - I sure don't. But zero works.

Further IPv6 differences

- IPv6 has a neat feature. Each host has a fixed fe80::/64 Link Local address which can be calculated based on its MAC address
- However, since a single MAC address might be present on multiple segments and interfaces, we can't connect unambiguously to fe80::/64 addresses
- So.. the full address is fe80::92fb:a6ff:fe4a:51da%eth0
- To this day, many browers can't connect to such addresses (including Chrome)

Presentation format

- IPv4 easy: 130.161.180.1:53, port 53 of 130.161.180.1
- ":" is used in IPv6 already, would be ambiguous
- Ways:
  a. [fe80::92fb:a6ff:fe4a:51da%eth0]:53
  b. fe80::92fb:a6ff:fe4a:51da%eth0#53
  c. fe80::92fb:a6ff:fe4a:51da%eth0@53
- I recommend 'a', but in any case **PICK ONE** both for input and output!

Conversion

- getaddrinfo() is **_the only way_** to convert "presentation format" IPv4 and IPv6 addresses. **Ignore anything else**.
  - Do **NOT** ignore the nonstandard way getaddrinfo() returns error codes!
  - Also, on some hosts, getaddrinfo() refuses to do IPv6 even for [::]!
- getaddrinfo() does **not** deal with [::]:25 notations for you.
  - Does deal with scoped addresses using the local convention

Policy: what does a host name mean?

- xs.powerdns.com has an IPv4 address and an IPv6 address
- If someone specifies: remote = "xs.powerdns.com", what do they mean?
  - Try IPv6 first, if that times out/generates error, try IPv4
    - And cache this <- probably best, but ...
  - Try either IPv4 or IPv6 first ("don't care")
  - Please don't try IPv6, it is slower and I did not know xs.powerdns.com had IPv6!!
    - "likely"
- Allow some way to be explicit, but still name based!

Filters, access control lists

- Easy to forget. I once turned my postfix into a global spam relay this way with an undercooked patch
  - World discovered while I was on holiday. Unhappiness ensued
- Make sure your users can filter on IPv4 *and* IPv6 ACLs
- Make sure your IPv6 ACL does not pass/block IPv4 traffic and vice versa
  - If you block 130.161.0.0/16 but after upgrade this looks like ::ffff:130.161.252.29...
    - oops

Proxies, webservers, forwarders...

- So once you think you are done with IPv6, people discover all those IPv4 only features you've been adding over the past decade
- For your checklist:
  - Can you **forward** things? Double check you can forward them to IPv6
  - Does your process have a built in webserver? Don't blindly bind it to 0.0.0.0:8080!
  - Do you accept stuff from proxies and parse headers? Make sure you parse them for IPv6 too!
  - Use libraries to connect to backends? Check their IPv6 resolution policy, if it is different from yours, document!

Some thoughts on statistics

- Common to maintain statistics based on remote IPv4 addresses
- Most people own 0.5 IPv4 address or so, need giant botnet to really generate traffic from a lot of distinct IPv4 addresses
- Even 'lite' users control 18446744073709551616 IPv6 addresses
- **If you keep any kind of state per client IP address, IPv6 will allow anyone to exhaust your memory**

Finally..

- "rgrep -E AF_INET[^6] ."
  - All lines without 'if' or part of socket() calls need to be audited
  - In general, try to minimize the number of times that you are explicit about AF_INET/AF_INET6!
- For fun, remove your IPv4 loopback interface and see if your software functions
- http://bert-hubert.blogspot.nl/2012/08/a-few-quick-notes-on-making-application.html
- http://blog.netherlabs.nl/articles/2006/10/12/the-joys-of-mixing-c-and-c

Making an application fully IPv6 compliant

A modest check list

http://tinyurl.com/ipv6-checklist

bert.hubert@powerdns.com
+31-622-440-095

POWERDNS