Ruby master - Bug #13593
Addrinfo#== behaves oddly
05/23/2017 03:30 PM - ioquatix (Samuel Williams)

Status: Rejected
Priority: Normal
Assignee:
Target version:
ruby -v:
Backport: 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description
It appears as if Addrinfo is using object identity. Addrinfo#== doesn't appear to work as one might expect, given the same instantiation:

```ruby
> irb
2.4.0 :001 > require 'socket'
=> true
2.4.0 :002 > Addrinfo.tcp('0.0.0.0', 1234) == Addrinfo.tcp('0.0.0.0', 1234)
=> false
2.4.0 :003 > a = Addrinfo.tcp('0.0.0.0', 1234)
2.4.0 :004 > a == a
=> true
```

History
#1 - 05/24/2017 03:00 AM - shyouhei (Shyouhei Urabe)
Accurate observation. It does not define equality so falls back to Object's definition, which compares identity.

I guess this is not by design; just no one had seriously needed yet.

#2 - 06/02/2017 04:40 PM - valerauko (Balint Erdos)
- File addrinfo_eql.diff added

I wonder if naive comparison like this is advisable?

#3 - 07/14/2017 09:17 AM - akr (Akira Tanaka)
- Status changed from Open to Feedback

It is difficult compare Addrinfo objects precisely if not impossible.

Using Addrinfo#inspect as addrinfo_eql.diff is fragile.

Addrinfo is basically struct addinfo which contains struct sockaddr. Byte-to-byte comparison may not work. struct sockaddr can have padding which makes byte-to-byte comparison different but means same address. Also, ai_family may be AF_UNSPEC which is same meaning to AF_INET or AF_INET6. I guess there are more concerns.

The critical point is that Ruby can not know all protocols supported by OS. Since struct sockaddr depends to a protocol, we can not compare all socket address properly.

#4 - 10/21/2017 09:57 AM - akr (Akira Tanaka)
- Status changed from Feedback to Rejected

Precise definition of comparing Addrinfo is very difficult if not impossible

#5 - 08/27/2018 04:33 AM - ioquatix (Samuel Williams)
Doing a binary comparison would be acceptable.

I don't think so. As Akira already pointed out,

- Addrinfo have "padding" bits which are not stable. Binary comparison might fail at that point.
- Addrinfo can handle arbitrary socket addresses, not only internet things. Which makes it impossible to tell which bit is a padding and which is not.

PR here: https://github.com/ruby/ruby/pull/2259

What's the use case of this comparison?

We discussed about this in developer meeting. This feature has some difficulty in spec for example padding, unknown type (other than AF_INET, AF_INET6, AF_UNIX, and so on), which should it use for the comparison. Though such difficulty, we can define a spec only for AF_INET and AF_INET6 if there's concrete use case.