

erlang / otp Public[Code](#) [Issues](#) 371 [Pull requests](#) 155 [Actions](#) [Projects](#) [Wiki](#) [Settings](#)

Commit b057a9d

 **RaimoNiskanen** committed 3 weeks ago Verified

Update after feedback

- * Hide the randomized port number -1 internal feature better, so that it does not show in the type spec of `gen_udp:open/2`. Suppress the Dialyzer warnings that follows from that.
- * Mention RFC 5452 in the RFC list.
- * Explain in a code comment why we use a deprecated `crypto` function.
- * Fix `inet:bind_random` to actually do a final attempt with port 0. Reduce the number of random attempts to 3 since the "always" succeed anyway on a "normal" machine.
- * Fix SCTP modules to handle port number like the TCP and UDP modules.

[master](#) (#10864) · [OTP-28.4.2](#) [OTP-27.3.4.10](#)1 parent [b1e0f67](#) commit b057a9d **7 files changed** +58 -26 lines changed[↑ Top](#) 

v lib/kernel/src

- gen_udp.erl
- inet.erl
- inet6_sctp.erl
- inet_db.erl
- inet_dns.erl
- inet_res.erl
- inet_sctp.erl

7 files changed +58 -26 lines changed

Search within code



lib/kernel/src/gen_udp.erl



```

@@ -141,15 +141,15 @@
141 141 %% -- open -----
142 142
143 143 -spec open(Port) -> {ok, Socket} | {error, Reason} when
144 -   Port    :: inet:port_number() | -1,
144 +   Port    :: inet:port_number(),
145 145   Socket  :: socket(),
146 146   Reason  :: system_limit | inet:posix().
147 147
148 148 open(Port) ->
149 149   open(Port, []).
150 150
151 151 -spec open(Port, Opts) -> {ok, Socket} | {error, Reason} when
152 -   Port    :: inet:port_number() | -1,
152 +   Port    :: inet:port_number(),
153 153   Opts    :: [inet:inet_backend() | open_option()],
154 154   Socket  :: socket(),
155 155   Reason  :: system_limit | inet:posix().

```

lib/kernel/src/inet.erl



```

@@ -1785,7 +1785,7 @@ gethostbyaddr_tm_native(Addr, Timer, Opts) ->
1785 1785     {ip6_address() | 'any' | 'loopback',
1786 1786     port_number()}} |
1787 1787     undefined, % Internal - no bind()
1788 -   BPort  :: port_number() | -1,
1788 +   BPort  :: port_number(),
1789 1789   Opts   :: [socket_setopt()],
1790 1790   Protocol :: socket_protocol(),
1791 1791   Family  :: address_family(),
@@ -1840,7 +1840,7 @@ open(Fd_or_OpenOpts, BAddr, BPort, Opts, Protocol,
1840 1840     {ip6_address() | 'any' | 'loopback',
1841 1841     port_number()}} |
1842 1842     undefined, % Internal - translated to 'any'

```

1843	-	BPort :: port_number() -1,
1843	+	BPort :: port_number(),
1844	1844	Opts :: [socket_setopt()],
1845	1845	Protocol :: socket_protocol(),
1846	1846	Family :: address_family(),
⋮ ↓ ↑ ⋮		@@ -1944,24 +1944,38 @@ open_setopts(S, BAddr, BPort, Opts, Module) ->
1944	1944	
1945	1945	
1946	1946	
1947	-	bind(S, Addr, Port) when is_list(Addr), ?port(Port) ->
1947	+	bind(S, Addr, Port) when is_list(Addr), is_integer(Port) ->
1948	1948	bindx(S, Addr, Port);
1949	1949	bind(S, Addr, -1) ->
1950	-	bind_random(S, Addr, 5);
1951	-	bind(S, Addr, Port) when ?port(Port) ->
1952	-	%% ?DBG([s, S], {addr, Addr}, {port, Port}]],
1953	-	prim_inet:bind(S, Addr, Port).
1954	-	
1955	-	bind_random(S, Addr, Cnt) when is_integer(Cnt) ->
1950	+	bind_random(S, Addr);
1951	+	bind(S, Addr, Port) when is_integer(Port) ->
1952	+	do_bind(S, Addr, Port).
1953	+	
1954	+	do_bind(S, Addr, Port) ->
1955	+	Result = prim_inet:bind(S, Addr, Port),
1956	+	%% ?DBG([s, S], {addr, Addr}, {port, Port}, Result]],
1957	+	Result.
1958	+	
1959	+	bind_random(S, Addr) ->
1960	+	Cnt = 3,
1961	+	bind_random(S, Addr, Cnt).
1962	+	%%
1963	+	bind_random(S, Addr, 0 = _Cnt) ->
1964	+	Port = 0,
1965	+	do_bind(S, Addr, Port);
1966	+	bind_random(S, Addr, Cnt) when is_integer(Cnt), 0 < Cnt ->
1956	1967	Port = inet_db:res_option(random_port),
1957	-	%% ?DBG([s, S], {addr, Addr}, {port, Port}]],
1968	+	bind_random(S, Addr, Cnt, Port).

```

1969 + %%
1970 + bind_random(S, Addr, _Cnt, 0 = Port) ->
1971 +     do_bind(S, Addr, Port);
1972 + bind_random(S, Addr, Cnt, Port) when ?port(Port) ->
1958 1973     case prim_inet:bind(S, Addr, Port) of
1959         -     {ok, _} = OK           -> OK;
1960         -     {error, _} = Error ->
1961             -     if Port ==:= 0           -> Error;
1962                 -     0 < Cnt           -> bind_random(S, Addr, Cnt - 1);
1963                 -     true             -> Error
1964             -     end
1974 +     {ok, _} = OK ->
1975 +         %% ?DBG([s, S], {addr, Addr}, {port, Port}, OK]),
1976 +         OK;
1977 +     {error, _} ->
1978 +         bind_random(S, Addr, Cnt - 1)
1965 1979     end.
1966 1980
1967 1981     bindx(S, [Addr], Port0) ->

```

lib/kernel/src/inet6_sctp.erl

```

... @@ -1,7 +1,7 @@
1 1 %%
2 2 %% %CopyrightBegin%
3 3 %%
4 - %% Copyright Ericsson AB 2007-2022. All Rights Reserved.
4 + %% Copyright Ericsson AB 2007-2026. All Rights Reserved.
5 5 %%
6 6 %% Licensed under the Apache License, Version 2.0 (the "License");
7 7 %% you may not use this file except in compliance with the License.
... @@ -54,11 +54,12 @@ open(Opts) ->
54 54         ifaddr = Addr,
55 55         port   = Port,
56 56         type   = Type,
57 -         opts   = S0s}} ->
57 +         opts   = S0s}}
58 +         when ?port(Port) ->
58 59         inet:open_bind(

```

```

59 60          Fd, Addr, Port, S0s, ?PROTO, ?FAMILY, Type, ?MODULE);
60 - Error ->
61 - Error
61 + {ok, _} -> {error, badarg};
62 + Error -> Error
62 63      end.
63 64
64 65      close(S) ->

```

lib/kernel/src/inet_db.erl

```

@@ -2109,6 +2109,10 @@ generate_random_port() ->
2109 2109      undefined -> 0
2110 2110      end.
2111 2111
2112 + %% We use `crypto:rand_uniform/2` here, which is the simplest to use,
2113 + %% but it is deprecated, for outdated reasons. In really old, now obsolete,
2114 + %% libcrypto versions the function was not cryptographically strong,
2115 + %% but since OpenSSL 1.1.0 that is no longer the case.
2112 2116      -compile({nowarn_deprecated_function, {crypto,rand_uniform,2}}).
2113 2117
2114 2118      crypto_rand_range(Range) when is_integer(Range), 0 < Range ->

```

lib/kernel/src/inet_dns.erl

```

@@ -26,6 +26,7 @@
26 26      %% RFC 6891: Extension Mechanisms for DNS (EDNS0)
27 27      %% RFC 2782: A DNS RR for specifying the location of services (DNS SRV)
28 28      %% RFC 2915: The Naming Authority Pointer (NAPTR) DNS Resource Rec
29 + %% RFC 5452: Measures for Making DNS More Resilient against Forged Answers
29 30      %% RFC 6488: DNS Certification Authority Authorization (CAA) Resource Record
30 31      %% RFC 7553: The Uniform Resource Identifier (URI) DNS Resource Record
31 32      %% RFC 6762: Multicast DNS

```

lib/kernel/src/inet_res.erl

```

@@ -757,6 +757,11 @@ udp_connect(#sock{inet=I}, {A,B,C,D}=IP, Port,
Verbose)
757 757      when ?ip(A,B,C,D), ?port(Port), is_boolean(Verbose) ->

```

```

758 758     udp_connect_socket(I, IP, Port, Verbose).
759 759
760 + %% Port number -1 is an internal feature that randomizes
761 + %% the bind port more than an ephemeral port, but it is not allowed
762 + %% by the type spec for gen_udp:open/2, so we have to suppress
763 + %% the Dialyzer warnings that that causes here...
764 + -dialyzer({[no_return, no_fail_call], udp_connect_fam/4}).
760 765     udp_connect_fam(Fam, IP, Port, Verbose) ->
761 766         case gen_udp:open(-1, [{active,false},binary,Fam]) of
762 767             {ok, Socket} = OK ->
@@ -1079,7 +1084,12 @@ query_ns(S0, {Msg, Buffer}, IP, Port, Timer,
Retry, I,
1079 1084         end
1080 1085     end.
1081 1086
1082 -
1087 + %% See udp_connect_fam/4 above. The use of port number -1 causes
1088 + %% a secondary Dialyzer warning here since Dialyzer concludes
1089 + %% that S cannot be 'undefined' because that would cause udp_connect/4
1090 + %% to crash here, so `S := undefined` can never be `true`.
1091 + %% This is a suppression for that.
1092 + -dialyzer({no_match, query_udp/7}).
1083 1093     query_udp(_S, _Msg0, _Buffer, IP, Port, 0, Verbose) ->
1084 1094         ?verbose(Verbose, "No try UDP server : -p:~p (overdue)\n",
1085 1095             [IP,Port]),
@@ -54,10 +54,12 @@ open(Opts) ->
54 54         ifaddr = Addr,

```

lib/kernel/src/inet_sctp.erl

```

... @@ -1,7 +1,7 @@
1 1 %%
2 2 %% %CopyrightBegin%
3 3 %%
4 - %% Copyright Ericsson AB 2007-2022. All Rights Reserved.
4 + %% Copyright Ericsson AB 2007-2026. All Rights Reserved.
5 5 %%
6 6 %% Licensed under the Apache License, Version 2.0 (the "License");
7 7 %% you may not use this file except in compliance with the License.
@@ -54,10 +54,12 @@ open(Opts) ->
54 54         ifaddr = Addr,

```

```
55 55      port = Port,  
56 56      type = Type,  
57 -      opts = S0s}} ->  
57 +      opts = S0s}}  
58 +      when ?port(Port) ->  
58 59          inet:open_bind(  
59 60              Fd, Addr, Port, S0s, ?PROTO, ?FAMILY, Type, ?MODULE);  
60 -      Error -> Error  
61 +      {ok, _} -> {error, badarg};  
62 +      Error -> Error  
61 63      end.  
62 64  
63 65      close(S) ->
```



Comments 0



Please [sign in](#) to comment.