IoSocketInetInvalidArgument

On startup spamd dies with "IO::Socket::INET: Invalid argument"

Overview

This bug is seen from time to time and is currently tracked down via the Gentoo Bug 58122. The reason *why* it happens is yet unknown, but it seems to be caused by a broken Perl.

Symptoms

When you start spamd with debugging you will see something like this:

```
$ spamd -D
trying to connect to syslog/unix...
no error connecting to syslog/unix
logging enabled:
        facility: mail
        socket:   unix
        output:   syslog
creating INET socket:
        Listen: 128
        LocalAddr: 127.0.0.1
        LocalPort: 783
        Proto: 6
        ReuseAddr: 1
        Type: 1
Could not create INET socket on 127.0.0.1:783: Invalid argument (IO::Socket::INET: Invalid argument)
```

The output above is from SpamAssassin 3.0.0, previous versions are a bit less verbose.

The FAQ on the amavisd-new site notes:

On some operating systems the Perl module IO::Socket reports obscure error on unsuccessful connect() in non-blocking mode as: "Invalid argument" in place of the true cause: "Connection refused".
The remote service should be checked if available and accessible (firewall rules? network problems?).

Another possible reason for "Invalid argument" reported by IO::Socket when running within chroot jail is some network-related system file missing in a chroot jail, e.g. /etc/protocols, /etc/services, /etc/netconfig, /etc/resolv.conf, /etc/nsswitch.conf, /etc/hosts, /etc/host.conf – depending on the operating system in use.
System debugging tools (debugger, strace, truss) may help to find the cause.

Diagnosis

The Gentoo bug mentioned above lead to the following wrong bind(2) call:

```
bind(5, {sa_family=0x3837 /* AF_??? */, sa_data="3\0\0\0\0\0\0\0\0001\306\1\0"}, 3) = -1 EINVAL (Invalid argument)
```

That one should actually look like:

```
bind(5, {sa_family=AF_INET, sin_port=htons(783), sin_addr=inet_addr("127.0.0.1")}, 16) = -1
```

Both the contents of the second parameter, `struct sockaddr *my_addr`, and the value of the third, `socklen_t addrlen`, are completely bogus. In hex they read:

```
37 38 33 00 00 00 00 00 00 00 00 00 30 31 03 30 | 783.........01.0
36 01 00                                        | 6..
```

Note the number 783, which is the port we try to connect on (as a string). The characters 1 and 6 might be the Type and Proto values above. Conclusion: Brain... err... Memfart 🙂

Cure

An upgrade to Perl 5.8.4 helped in the case mentioned. Recompiling Perl might have helped, too.

(MalteStretz)