UNIX & LINUX

socat parent process killed by failed connection

Asked 2 years, 11 months ago Active 2 years, 11 months ago Viewed 1k times

2

Short Question

Failed connection attemps to a forking `socat` process seem to be killing the parent process. Is this a bug? What's the best solution to this?

Long Question

I'm running a `socat` server (socat versions 1.7.3.1 and 1.7.3.2) with OpenSSL encryption (for verification of the server). I'm using the following command the start the server:

```
socat -d -d -d -U \
-lf /var/log/socat.log \
openssl-listen:8888,fork,reuseaddr,cert=server.pem,dhparam=dhparam.pem,verify=0 \
gopen:"file.txt" &
```

I'm using the `fork` option, so that each connection to the server should run in a separate child process. I can successfully connect to the server using the following client command:

```
socat - openssl-connect:hostname:8888,cafile=server.crt
```

I also tried connecting without providing the server certificate, i.e.:

```
socat - openssl-connect:hostname:8888
```

When I did that, I got the following error (as expected):

```
YYYY/mm/dd HH:MM:SS socat[3464] E SSL_connect(): error:1416F086:SSL
routines:tls_process_server_certificate:certificate verify failed
```

But the server process also terminates. This was unexpected. I get similar results with other unsuccessful connection attempts, e.g. trying to connect with netcat:

```
nc hostname 8888
```

In this case, the connection hangs until I kill it. However, as before, this doesn't just kill the current connection attempt (the child process), it also kills the server (the parent process).

I tried checking the logfile. During the connection attempt, the following messages are written to the logfile:

```
YYYY/mm/dd HH:MM:SS socat[PPID] D select -> (, 0x40, 0x0, 0x0, NULL/0.000000), 1
YYYY/mm/dd HH:MM:SS socat[PPID] D accept(6, 0xbedc0224, 0xbedc020c)
YYYY/mm/dd HH:MM:SS socat[PPID] I accept(6, {2, AF=2 127.0.0.1:54862}, 16) -> 7
YYYY/mm/dd HH:MM:SS socat[PPID] D fcntl(7, 2, 1)
YYYY/mm/dd HH:MM:SS socat[PPID] D fcntl() -> 0
YYYY/mm/dd HH:MM:SS socat[PPID] D getpeername(7, 0xbedc0234, 0xbedc021c{16})
YYYY/mm/dd HH:MM:SS socat[PPID] D getpeername(, {AF=2 127.0.0.1:54862}, {16}) -> 0
YYYY/mm/dd HH:MM:SS socat[PPID] D getsockname(7, 0xbedc02a4, 0xbedc0220{16})
YYYY/mm/dd HH:MM:SS socat[PPID] D getsockname(, {AF=2 127.0.1.1:8888}, {16}) -> 0
YYYY/mm/dd HH:MM:SS socat[PPID] N accepting connection from AF=2 127.0.0.1:54862 on AF=2 127.0.1.1:8888
YYYY/mm/dd HH:MM:SS socat[PPID] I permitting connection from AF=2 127.0.0.1:54862
YYYY/mm/dd HH:MM:SS socat[PPID] D sigprocmask(0, 0xbedc0314, 0x0)
YYYY/mm/dd HH:MM:SS socat[PPID] D sigprocmask() -> 0
YYYY/mm/dd HH:MM:SS socat[PPID] D fork()
YYYY/mm/dd HH:MM:SS socat[PPID] D fork() -> PID
YYYY/mm/dd HH:MM:SS socat[PPID] N forked off child process PID
YYYY/mm/dd HH:MM:SS socat[PPID] I close(7)
YYYY/mm/dd HH:MM:SS socat[PPID] D close()  -> 0
YYYY/mm/dd HH:MM:SS socat[PPID] D sigprocmask(1, 0xbedc0314, 0x0)
YYYY/mm/dd HH:MM:SS socat[PPID] D sigprocmask() -> 0
YYYY/mm/dd HH:MM:SS socat[PPID] I still listening
YYYY/mm/dd HH:MM:SS socat[PPID] N listening on AF=2 0.0.0.0:8888
YYYY/mm/dd HH:MM:SS socat[PPID] D select(7, &0x48, &0x0, &0x0, NULL/0.000000)
YYYY/mm/dd HH:MM:SS socat[PID] D fork() -> 0
YYYY/mm/dd HH:MM:SS socat[PID] D getpid()
YYYY/mm/dd HH:MM:SS socat[PID] D getpid() -> PID
YYYY/mm/dd HH:MM:SS socat[PID] I just born: child process PID
YYYY/mm/dd HH:MM:SS socat[PID] D setenv("SOCAT_PID", "0", 1)
YYYY/mm/dd HH:MM:SS socat[PID] D setenv() -> 0
YYYY/mm/dd HH:MM:SS socat[PID] D getpid()
YYYY/mm/dd HH:MM:SS socat[PID] D getpid() -> PID
YYYY/mm/dd HH:MM:SS socat[PID] D sigprocmask(1, 0xbedc0314, 0x0)
YYYY/mm/dd HH:MM:SS socat[PID] D sigprocmask() -> 0
YYYY/mm/dd HH:MM:SS socat[PID] I just born: child process PID
YYYY/mm/dd HH:MM:SS socat[PID] D setenv("SOCAT_PID", "PID", 1)
YYYY/mm/dd HH:MM:SS socat[PID] D setenv() -> 0
```

So it looks like there's a segmentation fault (signal 11) in the child process, and this causes the parent process to terminate. My current work-around is to run the socat command in a loop, e.g.:

```
while true; do
socat -d -d -d -U \
-lf /var/log/socat.log \
openssl-listen:8888,fork,reuseaddr,cert=server.pem,dhparam=dhparam.pem,verify=0 \
gopen:"file.txt" &
done
```

But this seems inelegant - surely there must be a better solution.

linux networking openssl segmentation-fault socat

Share Improve this question

Follow

edited Oct 8 '18 at 20:55

asked Oct 8 '18 at 20:37

igal

8,514 ● 1 ● 34 ● 55