How to find the process binding a TCP port

How to find the process binding a TCP port?

We have several error scenarios on which some SAP process cannot start because some other process has already bound its TCP port.

```
*** ERROR => NiIBindSocket: SiBind failed for hdl 209/sock 35
        (SI_EPORT_INUSE/98; I4; ST; 0.0.0.0:8080) [nixxi.cpp 3827]
```

This error message that can be seen in most of the component trace files, tells us that port 8000 is already being used (bound) by another process. To find out which process is using this port, the `netstat` tool can be used. `Netstat` is a command-line tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface statistics. Follows the procedure example on both UNIX and WINDOWS platforms:

**LINUX:**

```
> netstat -nepal|grep 8080
tcp 0 0 :::8080 :::* LISTEN 10404 18906159 7952/java
```

In this first example (Linux) we see that process with PID 7952 (java) is binding TCP port 8080.

**WINDOWS:**

```
> netstat -nao|findstr 8080
TCP 0.0.0.0:8080 0.0.0.0:0 LISTENING 3696
TCP [:]8080 [:]0 LISTENING 3696
```

In the Windows example, we find out that process with PID 3696 is the one binding the relevant port (8080).

It is also possible to use `netstat -ab` in Windows where it will list the process responsible to bind the port. The option "a" list all connections and the option "b" display the executable involved.

**AIX:**

```
> netstat -Aan | grep 8000 | grep LISTEN
f1000e000130bbb8 tcp4 0 0 *.8000 *.
LISTEN
> rmsock f1000e000130bbb8 tcpcb
The socket 0xf1000e000130b808 is being held by process 13893704 (icman).
> ps -ef | grep 13893704
vmcadm 13893704 9175156 0 Dec 22 - 0:17 icman -attach
pf=/usr/sap/VMC/SYS/profile/VMC_DVEBMGS00_is4039
```

**SOLARIS:**
The first command will print a list with the PID of all processes that are active at the system. The port number must be adjusted at the first "grep" command.

The PID that is using the port will be the one immediately before the line that ends with "port: <port>" ("port: 3900" in the example above).

The second command will show information about the process itself.

With this information, it can be decided whether this process should use this port or whether it can be stopped or reconfigured to use another port.