

Re: Interface security considerations

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2007-01/msg00215.html>

- *From:* Robert Watson <rwatson@xxxxxxxxxxx>
 - *Date:* Wed, 31 Jan 2007 09:20:45 +0000 (GMT)
-

On Mon, 29 Jan 2007, Victor Loureiro Lima wrote:

– While I was trying to figure out which process was listening on a certain interfaces an idea for a patch in sys/net/bpf.c functions `bpf_open()` and `bpf_close()` that would simply `printf(9)` the PID (`d->bd_pid = td->td_proc->p_pid;`) of a process that was trying to open the BPF device, while it was a simple patch, I am not sure if using the BPF device is the only possible way to sniff the packets from an interface, I know that linux implement `sock_packet`, and some systems have DLPI, just to get things straight, If an application wants to be able to sniff packets on a interface the only possible way (without messing with kernel at all) is using the BPF interface or are there other ways (even if they are not portable out of FreeBSD at all) of doing this?

FYI, the semantics of `bd_pid` are potentially confusing: `bd_pid` is the process ID of the process that has most recently issued an operation on the BPF descriptor. Since more than one process at a time may have a reference to the same BPF descriptor (discouraged, but possible), `bd_pid` doesn't reflect all the processes that have in the past used the descriptor, or may even be using it when queried. `fstat` and other tools that derive consumer information from process and file descriptor tables are the only way to gather a full list of current consumers of a BPF descriptor.

BPF is the only formal API for network interface-layer packet sniffing, but there are a variety of services that can be used in FreeBSD to sniff packets. For example, using the IPFW firewall, it is possible to tee the packet stream to a divert socket, which provides access to the active packet stream visible to IP. PF and `ipfilter` also offer similar facilities, which I believe are even directly supported by `tcpdump`. And, the administrator can always load custom kernel code to the same effect, as there are network interface programming hooks in `netgraph` that allow attaching to the packet stream. Compiling BPF out of the kernel will make it more complex to sniff packets.

Robert N M Watson
Computer Laboratory
University of Cambridge

freebsd-net@xxxxxxxxxxx mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-net>
To unsubscribe, send any mail to "freebsd-net-unsubscribe@xxxxxxxxxxx"