filter.txt: Linux Socket Filtering
Written by: Jay Schulist <jschlst@samba.org>

Introduction

```
============
```

Linux Socket Filtering is derived from the Berkeley Packet Filter. There are some distinct differences between the BSD and Linux Kernel Filtering.

Linux Socket Filtering (LSF) allows a user-space program to attach a filter onto any socket and allow or disallow certain types of data to come through the socket. LSF follows exactly the same filter code structure as the BSD Berkeley Packet Filter (BPF), so referring to the BSD bpf.4 manpage is very helpful in creating filters.

LSF is much simpler than BPF. One does not have to worry about devices or anything like that. You simply create your filter code, send it to the kernel via the SO_ATTACH_FILTER ioctl and if your filter code passes the kernel check on it, you then immediately begin filtering data on that socket.

You can also detach filters from your socket via the SO_DETACH_FILTER ioctl. This will probably not be used much since when you close a socket that has a filter on it the filter is automagically removed. The other less common case may be adding a different filter on the same socket where you had another filter that is still running: the kernel takes care of removing the old one and placing your new one in its place, assuming your filter has passed the checks, otherwise if it fails the old filter will remain on that socket.

Examples

```
========
```

Ioctls-

```
setsockopt(sockfd, SOL_SOCKET, SO_ATTACH_FILTER, &Filter, sizeof(Filter));
setsockopt(sockfd, SOL_SOCKET, SO_DETACH_FILTER, &value, sizeof(value));
```

See the BSD bpf.4 manpage and the BSD Packet Filter paper written by Steven McCanne and Van Jacobson of Lawrence Berkeley Laboratory.