More details on the ss command in Linux

Taz Wake | Halkyn Consulting Ltd | Originally published 2023-11-15

A previous article mentioned using the ss (socket statistics) command instead of netstat when checking traffic on a Linux device. This resulted in a discussion of *why* people should change.

Really there are two reasons:

1. The netstat command is being deprecated.
2. The ss command is technically better. This is the bit I want to elaborate on here.

The (arguable) technical superiority of over primarily lies in how these tools interact with the Linux kernel to retrieve information about sockets and network connections. A lot of this is definitely subjective.

Before we begin this, please keep in mind that as of today, you absolutely can simply choose to use whatever tool you like.

Why is ss better?

Kernel Interaction and Performance

- **Netlink Interface (ss) vs. /proc File System (netstat)**: uses the Netlink socket API for communication with the Linux kernel. This approach is more efficient than 's method of parsing the filesystem (specifically , , etc.). For instance, when dealing with a large number of connections, can be slow because it reads and processes each line in the files, while can quickly fetch bulk data through the Netlink interface.

Detailed and Real-Time Data

- **More Extensive Data**: is capable of showing more detailed information about TCP and state information, which may not display. For example, can show TCP internal information like TCP retransmit queues, memory usage by each socket, and more detailed state information of the TCP connections.

Advanced Filtering Options

- **Enhanced Filtering**: provides advanced filtering options that allow users to get precise information. For instance, with , you can filter output based on state (`established`, , etc.), address families, or even cgroup (Control Groups). Such refined filtering isn't as straightforward with .

Examples Demonstrating Superiority

- **Performance Example**: When checking for open ports on a system with thousands of connections, would execute and return results significantly faster than , especially noticeable in high-load scenarios.

- **Detail Example**: To see detailed TCP information, provides extended insights, like showing the congestion control algorithm used by each TCP connection, something not available with .

- **Filtering Example**: With , you can filter sockets by cgroup using , which is a level of granularity not available in .

Summary

Really, is technically superior primarily because of its efficient kernel interaction, ability to provide more detailed and real-time data, and advanced filtering options. These capabilities make a more suitable tool for in-depth network analysis and diagnostics, especially in environments with extensive network activity.

Will you notice the difference between the two on (say) a VirtualMachine with under 100 network connections? Probably not. But it is still a deprecated tool.

About Halkyn Consulting

Halkyn Consulting Ltd is a UK-based security and risk management consultancy providing 24/7 incident response, digital forensics, threat hunting, virtual security leadership (vCISO) and security compliance support to organisations worldwide. We deliver clear, practical, vendor-independent advice based on real-world threats — no hype, no upsell.

More free security resources are available at www.halkynconsulting.co.uk/security-resources.

Web: www.halkynconsulting.co.uk | **Email:** info@halkynconsulting.co.uk | **Tel:** +44 1244 940858