Abstract

There has been an explosive increase in the number of private IP addresses that a single web application can contain due to the proliferation of container technologies like Docker and container orchestration systems such as Kubernetes, an open-source software. The application layer in the Web 3-tier architecture consists of multiple nodes linked to load balancers, with multiple containers riding on each node. Containers communicate across nodes via virtual network interfaces assigned IP addresses in CIDR-allocated IP ranges in the virtual network space built across nodes. Container orchestration systems manage private IP addresses for tens of thousands of containers in an extensive application.

In this background, a virtual network space within physical nodes is divided using cgroups, a Linux kernel function. To prevent network threats, packet filtering must be implemented between physical nodes and containers—some Container Network Interfaces (CNIs) support packet filtering between containers. There is a long precedent for using iptables for packet filtering in Linux. Still, it is a performance bottleneck in cases with many filtering rules (tens of thousands), as in the current clustering situation.

The Berkeley Packet Filter (BPF) tool has newly begun to be used for packet filtering as a substitute. Though BPF has been used as a packet filtering tool for a long time, it has achieved significant progress recently. Its versatility and ease of implementation have attracted attention for packet filtering and in various other areas. This study investigates the validity of the shift in packet filtering technology by examining iptables and BPF from various perspectives.