

SOLARIS 10 VS. PITBULL

There are two ways in which Solaris 10 differs from earlier Solaris version as relating to the Argus PitBull products: *privileges* and *zones*.

Privileges

Solaris 10 includes a “standard” least privilege (LP) mechanism that was required for all trusted systems built to meet the 1991 U.S. CMW criteria. This is essentially the same privilege and role mechanism that has been found in Trusted Solaris since its earliest days. The Solaris LP mechanism enforces a finer-grained administrative control than the standard Unix superuser/root mechanism.

Privileges: Solaris 10 vs. PitBull Foundation

The SOL10 LP mechanism is roughly equivalent to the Pbfd LP in terms of its general approach. Both have the idea of three basic privilege sets: Pbfd calls them effective, maximum, and limiting sets; SOL10 calls them effective, permitted, and limit sets. Both allow processes to “pick up” privileges when a binary file is executed, but there is a difference between how these work in each system. The SOL10 approach is to identify with a process what may be “inherited” from the exec; the Pbfd approach is to identify “inheritable” as a separate privilege set on the file. In these ways, neither SOL10 nor Pbfd seems to be inherently stronger or more flexible.

However, the Pbfd LP mechanism is more powerful and flexible than SOL10 in a number of ways. First, the Pbfd privileges are much more granular. SOL10 has fewer than 50 privileges, Pbfd has more than 100. This additional granularity can be important for better controlling processes that need security-override capability. Second, the Pbfd mechanism includes kernel functionality and system utilities to track and report on which privileges a process attempts to use. This “learn mode” feature significantly improves the ability for an off-the-shelf software package to be incorporated into a trusted environment. Third, the Pbfd privilege subsystem includes a variety of advanced superuser-emulation components, again improving compatibility with off-the-shelf software and significantly improving the ease of integrating software written for standard Unix’s superuser philosophy of administration.

Privileges: Solaris 10 vs. PitBull LX

The technical approach to controlling superuser is fundamentally different in SOL10 and PBLX. SOL10 is using a system-wide least privilege concept; PBLX is based on limiting the power of specific programs and processes. The SOL10 approach is to grant a limited set of capabilities to processes that need to perform privileged operations; the PBLX approach is to prevent a specified process from using certain superuser powers if the process is running as root. Both approaches solve the general problem of how to put a process into a privileged state that is less powerful (and therefore less dangerous) than the standard Unix superuser.

The advantage of the SOL10 approach is its granularity, which is finer-grained than the LX approach. The PBLX advantage is its absolute compatibility with the base operating system. Under PBLX the system is completely unchanged, with superuser/root being exactly as it has always been except for those programs and processes that are under PBLX control.

Zones

In SOL10, zones are a way to provide an isolated virtual machine. When a zone is created, the software packages needed by users and programs that will be operating in that zone must be “installed” into the zone. A zone’s file system is really just a subtree in the standard file system, and is very much like a chroot environment from standard Unix. A program or user in a zone can only access files, processes, devices, and network resources that are part of the same zone. Privileges are applicable only for resources within the zone. A zone can be given its own network address, and the only communication between zones is through network connections—just as if the zones were separate physical machines. There is only one copy of the operating system, however, and all programs running in all zones are using the same operating system and operating system resources.

The standard view of the system is seen as a special zone, called the global zone. When a process or user is operating in this global zone, all other zones appear as simply part of the larger file system structure. Users and processes operating in the global zone have only the standard Unix/Solaris constraints against interacting with processes and resources within zones. In essence, there is no “zone security” imposed on users or processes operating in the global zone.

Zones: Solaris 10 vs. PitBull Foundation

SOL10 zones provide slightly better virtualization than what is provided by PBF. Using a compartment in PBF with a chroot environment will be nearly identical to a zone, except that a SOL10 zone also isolates privilege usage within a zone. In addition, the zone utilities make it easier to set up and manage such virtual machines.

The advantage that PBF compartmentalization has over zones is that a PBF compartment can itself be subdivided, and compartments can be made to overlap, form a hierarchy, or share resources in ways other than through network connections. This additional flexibility allows much more sophisticated, powerful, and efficient architectures to be built compared to fully-isolated virtual machines. The networking component of PBF is also significantly more sophisticated, allowing isolation and/or sharing of individual resources such as port numbers, IP address, network interfaces, subnets, and protocols.

Zones: Solaris 10 vs. PitBull LX

The comparison between zones and PBLX domains is similar to that between zones and PBF. PBLX domains are intended to be highly flexible. PBLX domains are slightly less isolated than PBF compartments (and therefore SOL10 zones), but PBLX domains are significantly more flexible than PBF compartments (and therefore SOL10 zones). Read,

write, and execute access between PBLX domains are completely site-configurable, and there is no restriction imposed on the relationships between domains or between access modes. The PBLX networking capability is also much less rigid, allowing a site to determine exactly what network resources can be shared between domains.

PitBull Features Not in Solaris 10

Not addressed in this discussion are features of PBLX that simply have no corresponding SOL10 mechanism. These would include: mandatory access control, labeled printing, trusted computing base protection, user clearances, virtualized shared directories, and login enhancement mechanisms.

Summary

PitBull Foundation is more powerful and flexible than the new SOL10 security features. SOL10 zones are a better choice if the goal is to create virtual machines, but PBLX is a better choice for an environment where flexibility is important or where multiple applications need to be run in a secure yet interconnected mode.

PitBull LX and SOL10 are addressing fundamentally different issues. PBLX is significantly more simple, flexible, and compatible than SOL10. PBLX is not designed for creating virtual machines, and SOL10 is significantly better than PBLX if having isolated, virtual machines is a requirement. If the goal is to protect a system from a number of subsystems, programs, or services, PBLX can do so with the strength of the SOL10 mechanisms but with much less intrusion and in a much more intuitive way.

Version 1.2, 06/02/22