proc(5) File Formats Manual proc(5)

NAME

proc - process information, system information, and sysctl pseudo-filesystem

DESCRIPTION

The proc filesystem is a pseudo-filesystem which provides an interface to kernel data structures. It is commonly mounted at /proc. Typi? cally, it is mounted automatically by the system, but it can also be mounted manually using a command such as:

```
mount -t proc proc /proc
```

Most of the files in the proc filesystem are read-only, but some files are writable, allowing kernel variables to be changed.

Mount options

The proc filesystem supports the following mount options:

hidepid=n (since Linux 3.3)

This option controls who can access the information in /proc/pid directories. The argument, n, is one of the following values:

traditional behavior, and the default if this mount option is not specified.

1 Users may not access files and subdirectories inside any /proc/pid directories but their own (the /proc/pid directo? ries themselves remain visible). Sensitive files such as /proc/pid/cmdline and /proc/pid/status are now protected against other users. This makes it impossible to learn whether any user is running a specific program (so long as the program doesn't otherwise reveal itself by its behav? ior).

2 As for mode 1, but in addition the /proc/pid directories be? longing to other users become invisible. This means that /proc/pid entries can no longer be used to discover the PIDs on the system. This doesn't hide the fact that a process with a specific PID value exists (it can be learned by other means, for example, by "kill -0 $PID"), but it hides a process's UID and GID, which could otherwise be learned by employing stat(2) on a /proc/pid directory. This greatly complicates an attacker's task of gathering information about running processes (e.g., discovering whether some dae? mon is running with elevated privileges, whether another user is running some sensitive program, whether other users

gid=gid (since Linux 3.3)

Specifies the ID of a group whose members are authorized to learn process information otherwise prohibited by hidepid (i.e., users in this group behave as though /proc was mounted with hidepid=0). This group should be used instead of approaches such as putting nonroot users into the sudoers(5) file.

Overview

Underneath /proc, there are the following general groups of files and subdirectories:

/proc/pid subdirectories

Each one of these subdirectories contains files and subdirecto? ries exposing information about the process with the correspond? ing process ID.

Underneath each of the /proc/pid directories, a task subdirec? tory contains subdirectories of the form task/tid, which contain corresponding information about each of the threads in the process, where tid is the kernel thread ID of the thread.

The /proc/pid subdirectories are visible when iterating through /proc with getdents(2) (and thus are visible when one uses ls(1)

/proc/tid subdirectories

Each one of these subdirectories contains files and subdirecto? ries exposing information about the thread with the correspond? ing thread ID. The contents of these directories are the same as the corresponding /proc/pid/task/tid directories.

The /proc/tid subdirectories are not visible when iterating through /proc with getdents(2) (and thus are not visible when one uses ls(1) to view the contents of /proc).

/proc/self

When a process accesses this magic symbolic link, it resolves to the process's own /proc/pid directory.

/proc/thread-self

When a thread accesses this magic symbolic link, it resolves to the process's own /proc/self/task/tid directory.

/proc/[a-z]*

Various other files and subdirectories under /proc expose sys? tem-wide information.

All of the above are described in more detail below.

NOTES

Many files contain strings (e.g., the environment and command line) that are in the internal format, with subfields terminated by null bytes ('\0'). When inspecting such files, you may find that the results are more readable if you use a command of the following form to display them:

```
$ cat file | tr '\000' '\n'
```

SEE ALSO

cat(1), dmesg(1), find(1), free(1), htop(1), init(1), ps(1), pstree(1), tr(1), uptime(1), chroot(2), mmap(2), readlink(2), syslog(2), slabinfo(5), sysfs(5), hier(7), namespaces(7), time(7), arp(8), hdparm(8), ifconfig(8), lsmod(8), lspci(8), mount(8), netstat(8), procinfo(8), route(8), sysctl(8)

The Linux kernel source files: Documentation/filesystems/proc.rst, Documentation/admin-guide/sysctl/fs.rst, Documentation/admin-guide/sysctl/kernel.rst, Documentation/admin-guide/sysctl/net.rst, and Documentation/admin-guide/sysctl/vm.rst.