# proc 目录在 ctf 中的应用 · HacKerQWQ's Studio

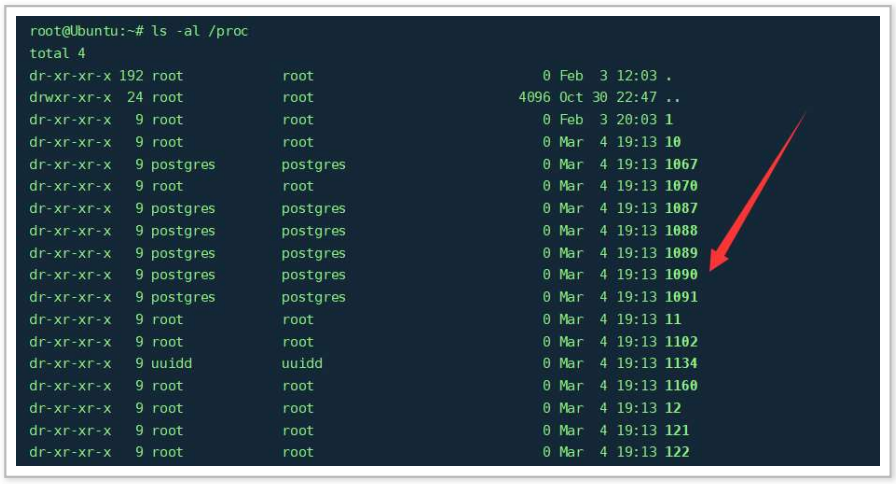
> /proc 目录 Linux 系统上的 / proc 目录是一种文件系统，即 proc 文件系统。

Linux 系统上的 / proc 目录是一种文件系统，即 proc 文件系统。与其它常见的文件系统不同的是，/proc 是一种伪文件系统（也即虚拟文件系统），存储的是当前内核运行状态的一系列特殊文件，用户可以通过这些文件查看有关系统硬件及 当前正在运行进程 的信息，甚至可以通过更改其中某些文件来改变内核的运行状态。

简单来讲，/proc 目录即保存在 系统内存 中的信息，大多数虚拟文件可以使用文件查看命令如 cat、more 或者 less 进行查看，有些文件信息表述的内容可以一目了然，但也有文件的信息却不怎么具有可读性。

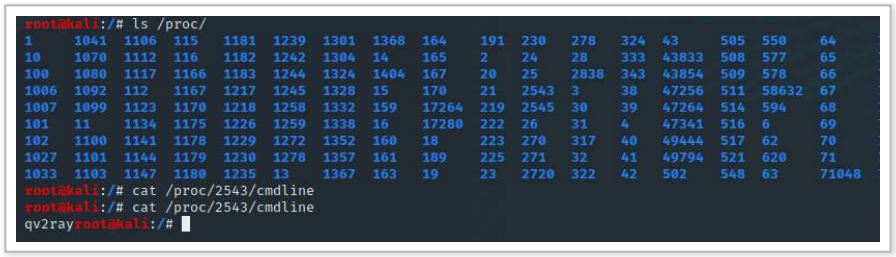
/proc 目录中包含许多以数字命名的子目录，这些数字表示系统当前正在运行进程的 进程号(PID) ，里面包含对应进程相关的多个信息文件：

```
ls -al /proc
```




cmdline 文件存储着启动当前进程的 完整命令 ，但僵尸进程目录中的此文件不包含任何信息。可以通过查看 cmdline 目录获取启动指定进程的完整命令：

```
cat /proc/2543/cmdline
```

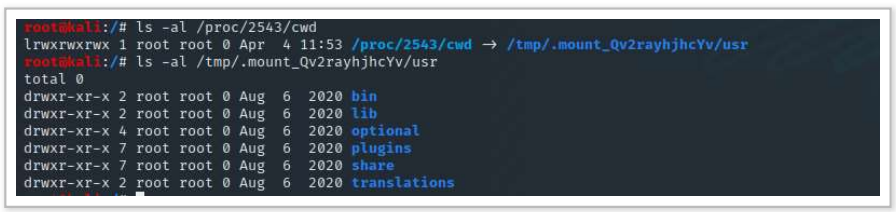



`

cwd 文件是一个指向 当前进程运行目录 的 符号链接 。可以通过查看 cwd 文件获取目标指定进程环境的 运行目录 ：

```
ls -al /proc/1090/cwd
```

效果是一样的



exe 是一个指向启动当前进程的 可执行文件（完整路径） 的 符号链接 。

```
ls -al /proc/2543/exe
```


environ 文件存储着当前进程的 环境变量 列表，彼此间用 空字符 （NULL）隔开。变量用大写字母表示，其值用小写字母表示。可以通过查看 environ 目录来获取 指定进程的环境变量 信息：

```
cat /proc/2543/environ
```

fd 是一个目录，里面包含这当前进程打开的每一个 文件的文件描述符 （file descriptor），这些文件描述符是指向实际文件的一个符号链接，即每个通过这个进程打开的文件都会显示在这里。所以我们可以通过 fd 目录里的文件获得指定进程打开的每个文件的路径以及文件内容。

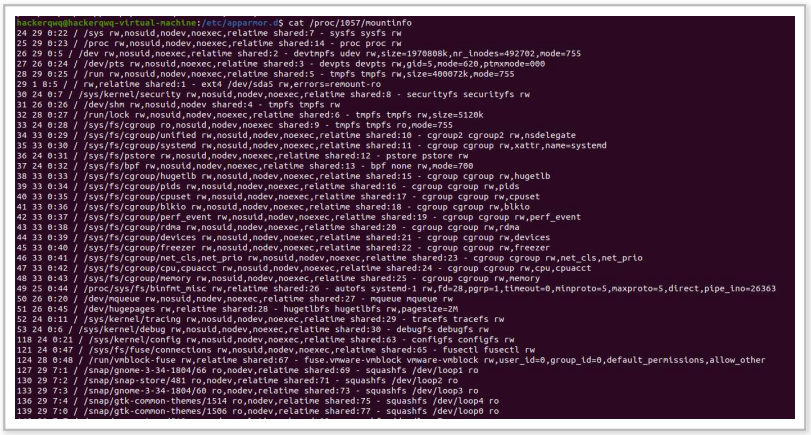
查看指定进程打开的某个文件的路径：

```
ls -al /proc/1070/fd
```

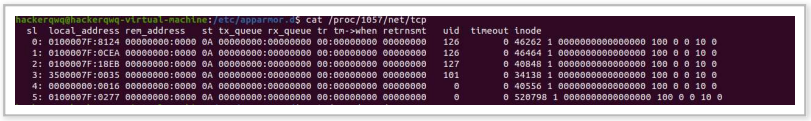
这个 fd 比较重要，因为在 linux 系统中，如果一个程序用 open() 打开了一个文件但最终没有关闭他，即便从外部（如 os.remove(SECRET_FILE)）删除这个文件之后，在 /proc 这个进程的 pid 目录下的 fd 文件描述符目录下还是会有这个文件的文件描述符，通过这个文件描述符我们即可得到被删除文件的内容。

指向进程的内存映射

指向进程所在文件系统挂载情况，常见 Docker 环境，此时 mounts 会泄露敏感路径




指向进程的网络信息，如读取 TCP 将获取进程所绑定的 TCP 端口，ARP 将泄露同网段内网 IP 信息




表示当前进程的 id

```
cat /proc/self/fd/id
```

读取本进程打开的文件

```
cat /proc/self/environ
```

读取本进程的环境变量

```
ls -al /proc/self/exe
```

读取本进程的可执行文件

```
ls -al /proc/self/cwd
```

读取本进程的程序执行目录

```
ls -al /prco/self/cmdline
```

读取本进程的执行的完整命令

```
cat /proc/self/mounts(mountinfo)
```

读取本进程的挂载情况

```
cat /proc/self/maps
```

读取本进程的内存映射情况

```
cat /proc/self/net/tcp(arp)
```

读取本进程的同进程绑定的 TCP 端口或内网 IP 信息

```
cat /proc/self/statm
```

包含了进程的内存使用信息

更多使用：
https://blog.csdn.net/mediatec/article/details/88578101

全文完