Autour du Buffer Overflow

CAP SETUID capability

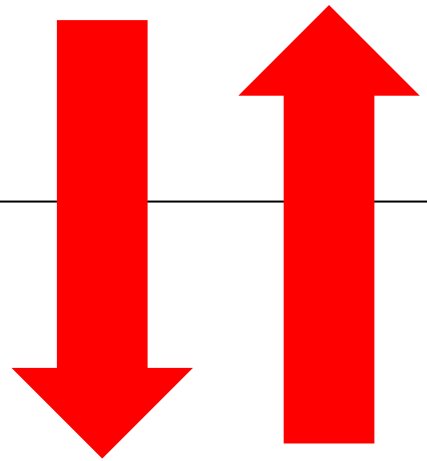
- For the purpose of performing permission checks, traditional UNIX implementations distinguish two categories of processes:
  - *privileged* processes (whose effective user ID is 0, referred to as superuser or root),
  - and *unprivileged* processes (whose effective UID is nonzero).
- The CAP_SETUID capability enables arbitrary manipulations of process UIDs
  - setuid(), seteuid()
- The seteuid() sets the effective user ID of the calling process. Unprivileged processes may only set the effective user ID to the real user ID, the effective user ID or the saved set-user-ID
- The setuid() function checks the effective user ID of the caller and if it is the superuser, all process-related user ID's are set to *uid*. After this has occurred, it is impossible for the program to regain root privileges.

Calling Process= shell (/bin/sh)
Called Process = sudo, su

non-root

seteuid()

root



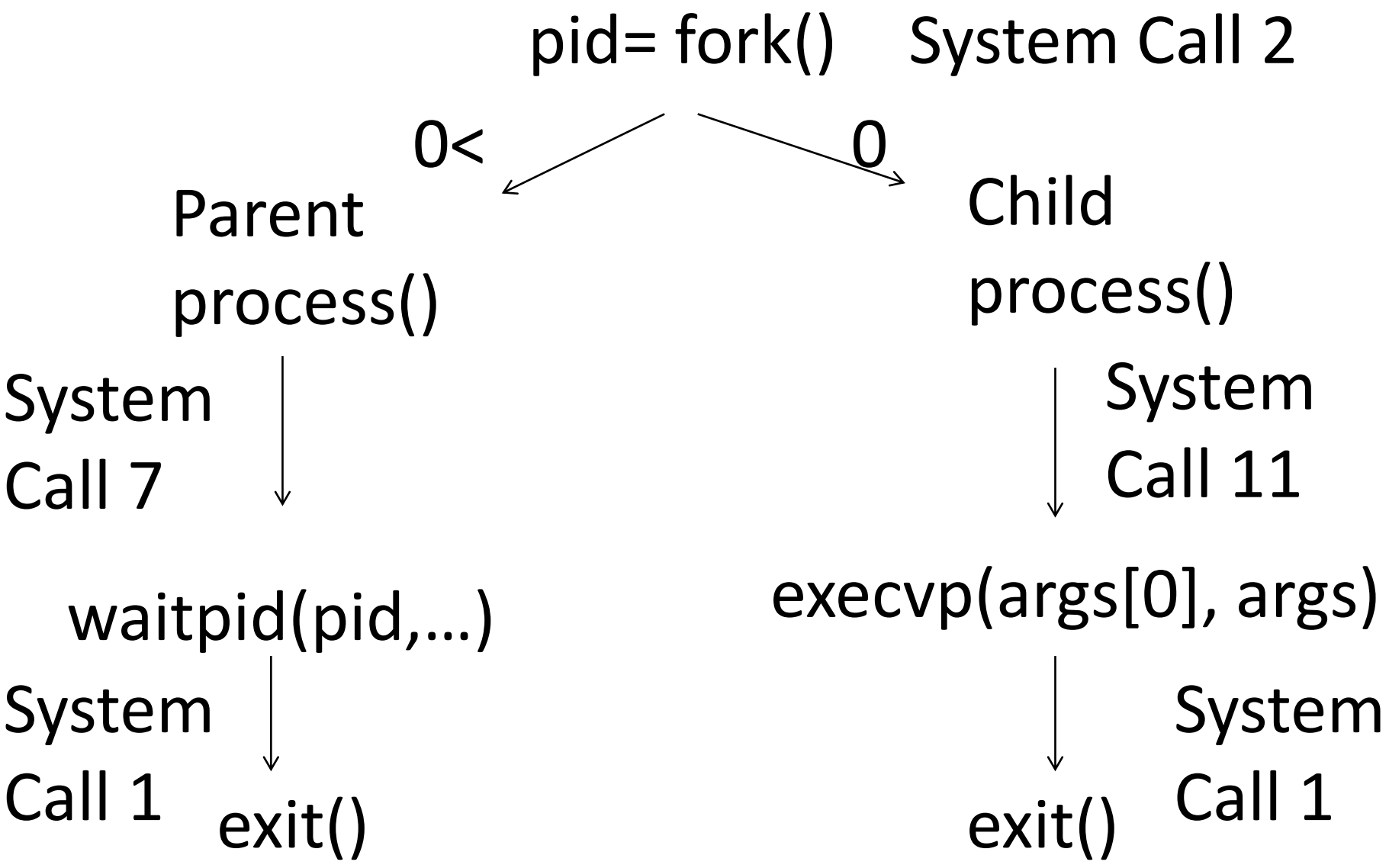
Process

PID	UID
pid	user
pid	0

setresuid(ruid, euid, suid) sets the real user ID, the effective user ID, and the saved set-user-ID of the calling process.

Linux: system call 164
mov eax, a4
mov ebx, 0 (ruid)
mov ecx,0 (euid)
mov edx,0 (suid)
Int 80

Shell : system()



SUDO.C

https://opensource.apple.com/source/sudo/sudo-9/sudo/sudo.c

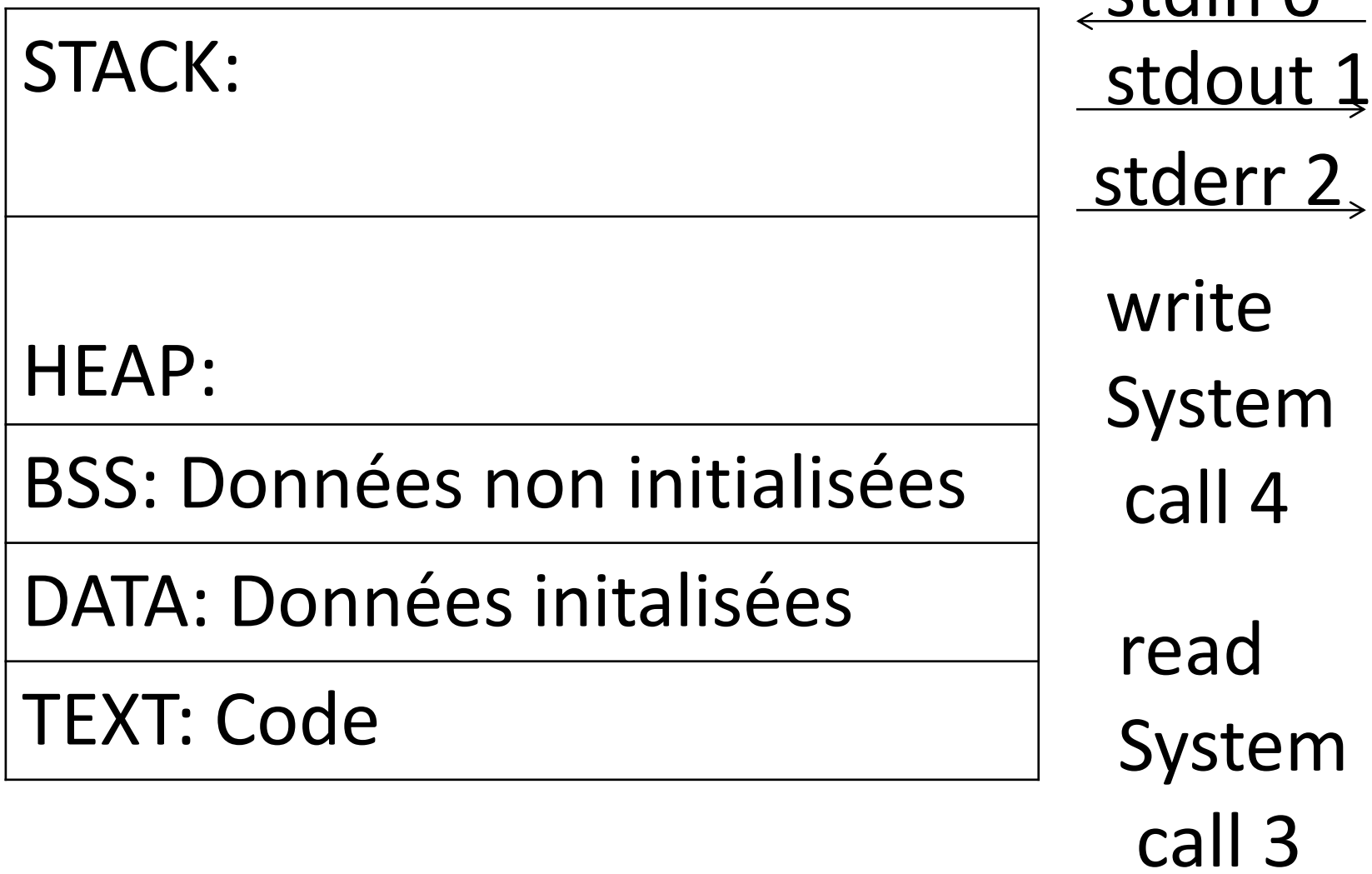
```
if (geteuid() != 0)
{ (void) fprintf(stderr, "Sorry, %s must be setuid root.\n", Argv[0]);
exit(1); }
```

```
setuid(0)
```

```
fork()
```

```
EXEC(safe_cmnd, NewArgv); /* run the command */
```

Modèle mémoire



Programme hello.c

Chargement
-Lecture fichier
-Allocation mémoire
-Transfert mémoire

-Exécution
call system 4

Hello
World

exit()
call system 1

call @proc

@ret -> SP

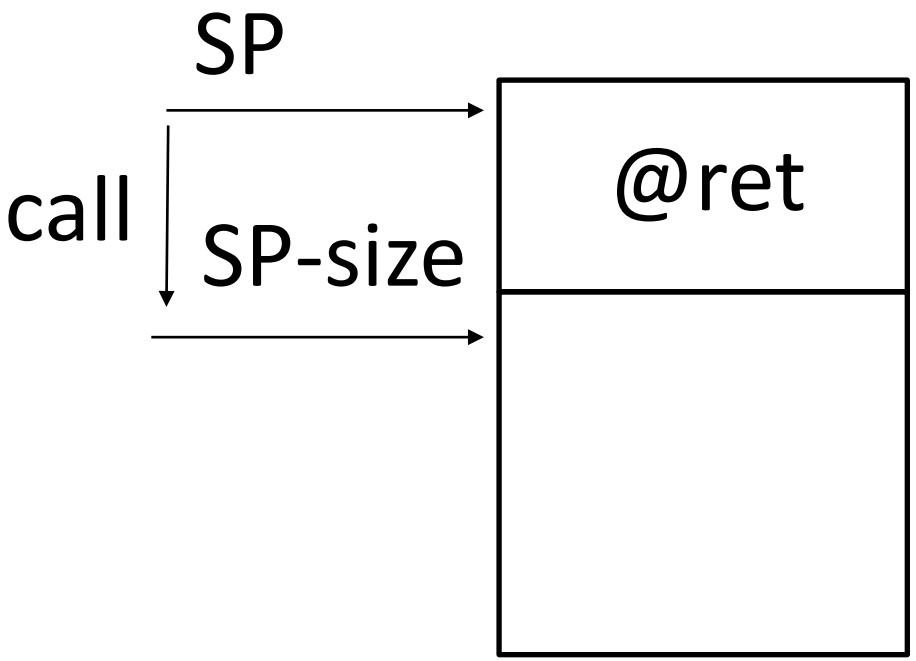
SP-size -> SP

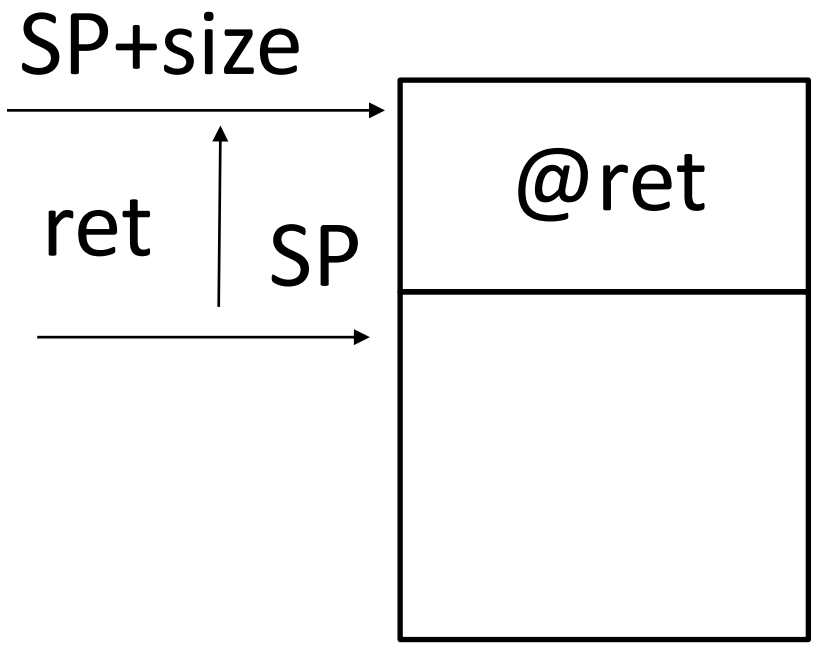
@proc->PC

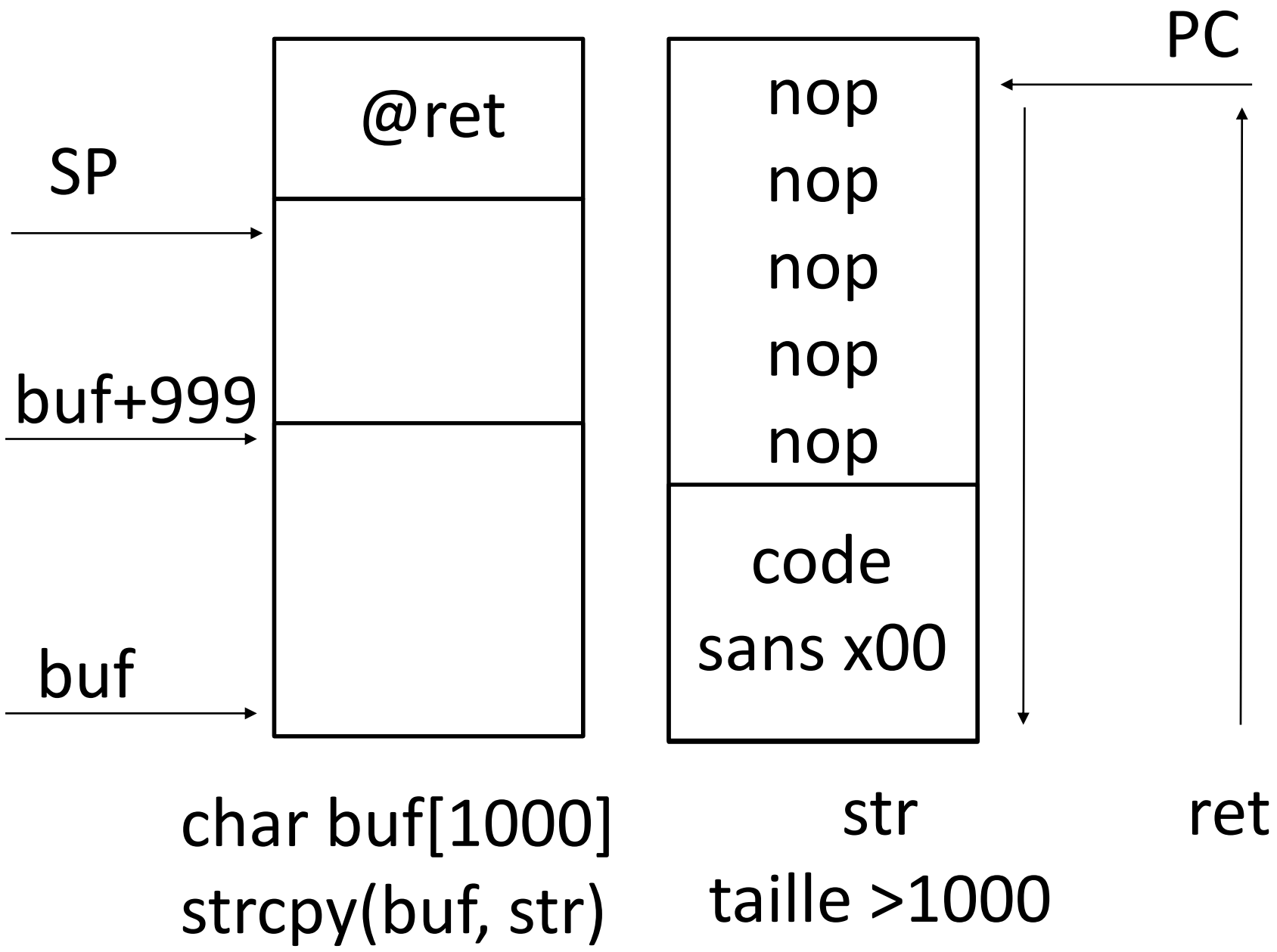
ret

SP+size -> SP

SP -> PC






PC
@ret
SP
buf+999
buf
nop
nop
nop
nop
nop
code
sans x00
char buf[1000]
strcpy(buf, str)
str
taille >1000
ret

Shell Code

- Fragment de code qui ne contient pas d’octet nul (0x00), et qui réalise un appel au Shell tel que execvp(" /bin/sh", args), avec optionnellement une escalade de privilège via setresuid.