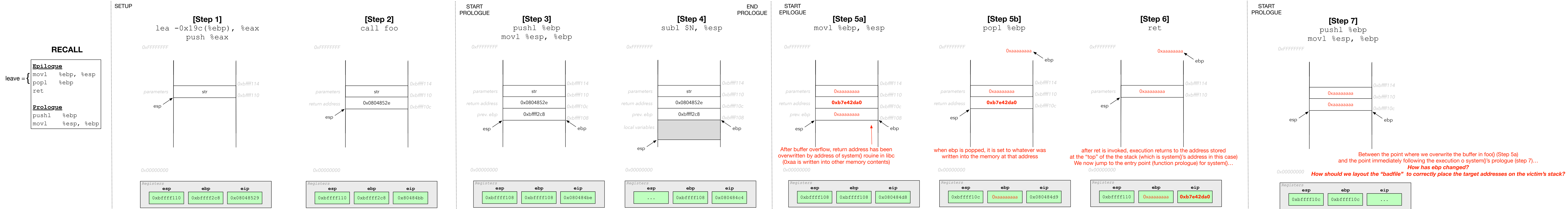


# Non-Conventional Enter/Leave Sequence

A Return-to-libc Attack via Buffer Overflow  
(Steps 1 - 4 are the same)



**SETUP**

**[Step 1]**

```
lea -0x19c(%ebp), %eax
push %eax
```

**[Step 2]**

```
call foo
```

**START PROLOGUE**

**[Step 3]**

```
pushl %ebp
movl %esp, %ebp
```

**END PROLOGUE**

**[Step 4]**

```
subl $N, %esp
```

**START EPILOGUE**

**[Step 5a]**

```
movl %ebp, %esp
```

**[Step 5b]**

```
popl %ebp
```

**[Step 6]**

```
ret
```

**START PROLOGUE**

**[Step 7]**

```
pushl %ebp
movl %esp, %ebp
```

0xFFFFFFFF

parameters

esp

str

0xbffff114

0xbffff110

0x00000000

Registers

esp	ebp	eip
0xbffff110	0xbffff2c8	0x08048529

0xFFFFFFFF

parameters

return address

esp

str

0xbffff114

0xbffff110

0x0804852e

0xbffff10c

0x00000000

Registers

esp	ebp	eip
0xbffff110	0xbffff2c8	0x80484bb

0xFFFFFFFF

parameters

return address

prev. ebp

esp

str

0xbffff114

0xbffff110

0x0804852e

0xbffff10c

0xbffff2c8

0xbffff108

0x00000000

Registers

esp	ebp	eip
0xbffff108	0xbffff108	0x080484be

0xFFFFFFFF

parameters

return address

prev. ebp

local variables

esp

str

0xbffff114

0xbffff110

0x0804852e

0xbffff10c

0xbffff2c8

0xbffff108

0x00000000

Registers

esp	ebp	eip
...	0xbffff108	0x080484c4

0xFFFFFFFF

parameters

return address

prev. ebp

esp

0xaaaaaaaa

0xb7e42da0

0xb7e42da0

0xb7e42da0

0xbffff114

0xbffff110

0xbffff10c

0xbffff108

0x00000000

Registers

esp	ebp	eip
0xbffff108	0xbffff108	0x080484d8

After buffer overflow, return address has been overwritten by address of system() routine in libc (0xaa is written into other memory contents)

0xFFFFFFFF

parameters

return address

esp

0xaaaaaaaa

0xb7e42da0

0xb7e42da0

0xb7e42da0

0xbffff114

0xbffff110

0xbffff10c

0xbffff108

0x00000000

Registers

esp	ebp	eip
0xbffff10c	0xaaaaaaaa	0x080484d9

when ebp is popped, it is set to whatever was written into the memory at that address

0xFFFFFFFF

parameters

return address

esp

0xaaaaaaaa

0xb7e42da0

0xb7e42da0

0xb7e42da0

0xbffff114

0xbffff110

0xbffff10c

0xbffff108

0x00000000

Registers

esp	ebp	eip
0xbffff110	0xaaaaaaaa	0xb7e42da0

after ret is invoked, execution returns to the address stored at the "top" of the the stack (which is system()'s address in this case)  
We now jump to the entry point (function prologue) for system()...

0xFFFFFFFF

parameters

return address

esp

0xaaaaaaaa

0xb7e42da0

0xb7e42da0

0xb7e42da0

0xbffff114

0xbffff110

0xbffff10c

0xbffff108

0x00000000

Registers

esp	ebp	eip
0xbffff10c	0xbffff10c	...

Between the point where we overwrite the buffer in foo() (Step 5a) and the point immediately following the execution o system()'s prologue (step 7)...

**How has ebp changed?**

**How should we layout the "badfile" to correctly place the target addresses on the victim's stack?**