Hello world shellcode

Hello world shellcode

```
section .data
string_msg:         db          ”Hello World”,0xa

section .text
global _start
_start:

        mov         rax,1
        mov         rdi,1
        mov         rsi,string_msg
        mov         rdx,12
        syscall
```

Hello world shellcode

```
section .data
string_msg:        db        ”Hello World”,0xa

section .text
global _start
_start:

        mov       rax,1
        mov       rdi,1
        mov       rsi,string_msg
        mov       rdx,12
        syscall
```

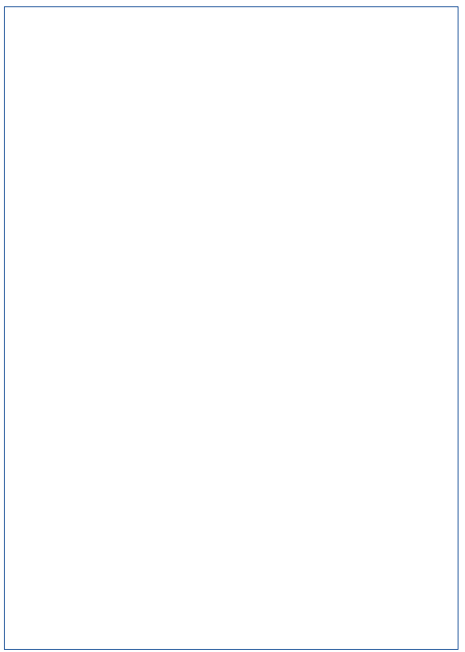
We cannot use hardcoded addresses of our hello world string

JMP CALL POP technique

We use jmp call pop technique which uses stack memory to load the addresse of our hello world string in our text section using pop instruction

```
section .text
global _start
_start:

        jmp one

shellcode:

        pop    rsi

        mov    rax,1
        mov    rdi,1
        mov    rdx,12
        syscall


one:

call            shellcode

string_msg: db      ”Hello world”,0xa
```

1



Stack memory

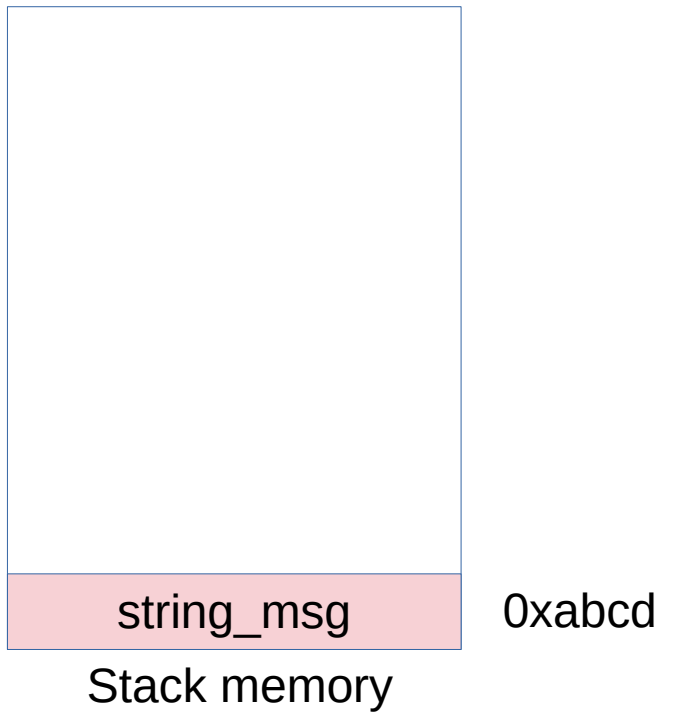
JMP CALL POP technique

We use jmp call pop technique which uses stack memory to load the addresse of our hello world string in our text section using pop instruction

```
section .text
global _start
_start:

        jmp one

shellcode:

        pop    rsi

        mov    rax,1
        mov    rdi,1
        mov    rdx,12
        syscall


one:

        call            shellcode

string_msg: db      ”Hello world”,0xa
```

1

2 Call stores the address of string_msg on stack before jumping to shellcode



JMP CALL POP technique

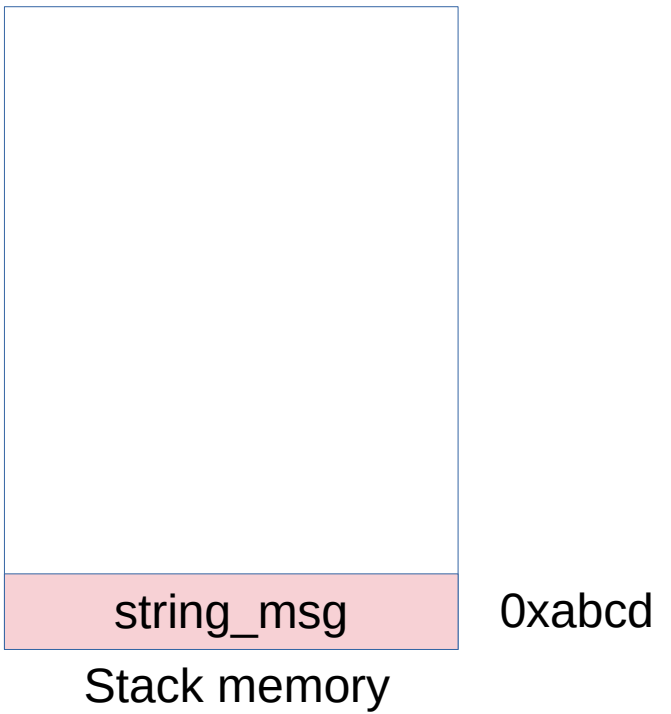
We use jmp call pop technique which uses stack memory to load the addresse of our hello world string in our text section using pop instruction

```
section .text
global _start
_start:

        jmp one

shellcode:

        pop    rsi

        mov    rax,1
        mov    rdi,1
        mov    rdx,12
        syscall

one:
        call   shellcode

string_msg: db      ”Hello world”,0xa
```

1

2 Call stores the address of string_msg on stack before jumping to shellcode

3



JMP CALL POP technique

We use jmp call pop technique which uses stack memory to load the addresse of our hello world string in our text section using pop instruction

```
section .text
global _start
_start:

    jmp one

shellcode:

    pop   rsi

    mov   rax,1
    mov   rdi,1
    mov   rdx,12
    syscall

one:

    call  shellcode

string_msg: db  "Hello world",0xa
```

1

2 Call stores the address of string_msg on stack before jumping to shellcode

3

4

string_msg address will be stored from the stack to rsi using pop