

DEP & ROP

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h

; __unwind {
    push     rbp
    mov     rbp, rsp
    push     rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_28]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     edx, [rax+0Ch]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     eax, ebx
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx

loc_30FB:                                     ; CODE XREF: Zei
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIc
    lea     rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt11
    mov     rdx, cs:_ZSt4endlIcSt11char_tra
    mov     rsi, rdx
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_318D
    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt11
    lea     rax, [rbp+var_14]
```

NX/DEP

- Giới thiệu **NX/DEP** (**No-Execute/Data Execution Prevention**)
 - Còn được biết đến là **Executable Space Protection**
 - Đừng nhầm lẫn với **\$esp** 😊
- Xuất hiện lần đầu trên Linux và Windows vào năm 2004.

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h

; __unwind {
    push     rbp
    mov     rbp, rsp
    push     rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     edx, [rax+0Ch]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+10h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     edx, ebx
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx

loc_30FB:                                     ; CODE XREF: Zei
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 0
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIc
    rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt13
    mov     rdx, cs:_ZSt4endlIcSt11char_tra
    mov     rsi, rdx
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_318D

    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt13
    lea     rax, [rbp+var_14]
```

DEP/NX/W^X

- Stack/Heap/data được map với No eXecute (chặn quyền thực thi)
- aka DEP (Data Execution Prevention)
- aka W^X (Write xor eXecute)
 - Phân vùng bộ nhớ chỉ được đánh dấu là **writable** HOẶC **executable** (không có cả hai).
 - Việc thực thi phân vùng bộ nhớ không được đánh dấu **executable** sẽ dẫn đến chương trình gặp **segmentation fault**.
- Không còn shellcode

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     ecx, [rbp+var_28]
    mov     eax, [ecx+0Ch]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     eax, ebx
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx
```

```
loc_30FB:
    mov     eax, [rbp+var_28]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout
    mov     rax, [rax]
    call    __ZStlsI11char_traitsIcESaIc>_ZStlsI11char_traitsIcESaIc>
    rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsI11char_traitsIcESaIc>_ZStlsI11char_traitsIcESaIc>
    mov     rdx, cs:_ZSt4endlIcSt11char_traitsIcESaIc>
    mov     rsi, rdx
    mov     rdi, rax
    call    __ZNSoIsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_31BD
    lea     rsi, aWannaCheatYes1 ; "wanna cheat?"
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsI11char_traitsIcESaIc>_ZStlsI11char_traitsIcESaIc>
    lea     rax, [rbp+var_14]
```

Runtime Memory

ELF Executable

.text

.rodata

.bss

Heap

Libraries (libc)

Stack

Runtime Memory without DEP

R-X

R-X

RW~~X~~

RW~~X~~

RW~~X~~

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = qword ptr -14h
;__unwind {
```

```
push rbp
mov rbp, rsp
push rbx
sub rsp, 28h
mov [rbp+var_28], rdi
mov [rbp+var_30], rsi
mov rax, [rbp+var_28]
mov eax, [rax+128h]
test eax, eax
jnz short loc_30FB
mov rax, [rbp+var_28]
edx, [rax+0Ch]
mov rax, [rbp+var_28]
ecx, [rax+14h]
mov rax, [rbp+var_30]
mov eax, [rax+10h]
mov ebx, ecx
sub ebx, eax
mov eax, ebx
add edx, eax
mov rax, [rbp+var_28]
mov [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
```

```
mov rax, [rbp+var_28]
mov eax, [rax+0Ch]
test eax, eax
jns loc_31C4
mov rax, [rbp+var_28]
add rax, 18h
mov rsi, rax
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsIcSt11char_traitsIcESaIc
rsi, aIsDead ; " is dead!"
mov rdi, rax
call __ZStlsISt11char_traitsIcEERSt13
rdx, cs:_ZSt4endlIcSt11char_tra
mov rsi, rdx
mov rdi, rax
call __ZNSolsEPFRSoS_E ; std::ostream
mov rax, [rbp+var_28]
mov eax, [rax+8]
test eax, eax
jnz short loc_31BD
lea rsi, aWannaCheatYes1 ; "wanna ch
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsISt11char_traitsIcEERSt13
lea rax, [rbp+var_14]
```

Runtime Memory

ELF Executable

.text

.rodata

.bss

Heap

Libraries (libc)

Stack

Runtime Memory without DEP

R-X

R--

RW-

RW-

RW-

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = qword ptr -14h
; __unwind {
```

```
    push     rbp
    mov      rbp, rsp
    push     rbx
    sub      rsp, 28h
    mov      [rbp+var_28], rdi
    mov      [rbp+var_30], rsi
    mov      rax, [rbp+var_28]
    mov      eax, [rax+128h]
    test     eax, eax
    jnz      short loc_30FB
    mov      rax, [rbp+var_28]
    mov      edx, [rax+0Ch]
    mov      rax, [rbp+var_28]
    mov      ecx, [rax+14h]
    mov      rax, [rbp+var_30]
    mov      eax, [rax+10h]
    mov      ebx, ecx
    sub      ebx, eax
    mov      eax, ebx
    add      edx, eax
    mov      rax, [rbp+var_28]
    mov      [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
```

```
    mov      rax, [rbp+var_28]
    mov      eax, [rax+0Ch]
    test     eax, eax
    jns      loc_31C4
    mov      rax, [rbp+var_28]
    add      rax, 18h
    mov      rsi, rax
    mov      rax, cs:_ZSt4cout_ptr
    mov      rdi, rax
    call     __ZStlsIcSt11char_traitsIcESaIc
    lea      rsi, aIsDead ; " is dead!"
    mov      rdi, rax
    call     __ZStlsISt11char_traitsIcEERSt13
    mov      rdx, cs:_ZSt4endlIcSt11char_tra
    mov      rsi, rdx
    mov      rdi, rax
    call     __ZNSolsEPFRSoS_E ; std::ostream
    mov      rax, [rbp+var_28]
    mov      eax, [rax+8]
    test     eax, eax
    jnz      short loc_31BD
    lea      rsi, aWannaCheatYes1 ; "wanna ch
    mov      rax, cs:_ZSt4cout_ptr
    mov      rdi, rax
    call     __ZStlsISt11char_traitsIcEERSt13
    lea      rax, [rbp+var_14]
```

Return Oriented Programming (ROP)

- Code reuse

- Không thể sử dụng code riêng vì không có buffer nào thực thi được
- Nhưng chúng ta vẫn có thể sử dụng những đoạn mã có sẵn của chương trình:

- ... điều gì sẽ xảy ra nếu chúng ta có thể kết hợp các đoạn code có sẵn để làm cho chương trình thực thi những gì ta muốn ?
- Những đoạn code đó được gọi là gadgets

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind[
push rbp
mov rbp, esp
push rbp
sub esp, 28h
mov [rbp+var_28], rdi
mov [rbp+var_30], rsi
mov rax, [rbp+var_20]
mov eax, [rax+128h]
test eax, eax
jnz short loc_30FB
mov rax, [rbp+var_28]
mov edx, [rax+0Ch]
mov rax, [rbp+var_28]
mov ecx, [rax+14h]
mov ebx, [rbp+var_14]
mov ebx, ecx
sub ebx, eax
add edx, eax
mov rax, [rbp+var_28]
mov [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
mov rax, [rbp+var_28]
mov [rax+0Ch], rax
jnz loc_31C4
mov rax, [rbp+var_28]
mov rax, [rbp+var_14]
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsIcSt11char_traitsIcESaIc
lea rsi, aIsDead ; " is dead!"
mov rdi, rax
call __ZStlsIcSt11char_traitsIcEERSt13
mov rdx, cs:_ZSt4endlIcSt11char_tra
mov rsi, rdx
mov rdi, rax
call __ZNSolsEPFRSoS_E ; std::ostream
mov rax, [rbp+var_28]
mov eax, [rax+8]
test eax, eax
jnz short loc_31BD
lea rsi, aWannaCheatYes1 ; "wanna ch
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsIcSt11char_traitsIcEERSt13
lea rax, [rbp+var_14]
```

Return Oriented Programming (ROP)

- ROP gadgets
 - Những đoạn code thường kết thúc với **return, syscall, call/jump**
 - Nếu chúng ta kiểm soát được **\$rsp – stack pointer**, chúng ta có thể kết hợp những gadgets với nhau tạo thành chuỗi được gọi là ROP gadgets.
 - Mỗi lần “**return**”, chương trình sẽ thực thi gadget tiếp theo ngay sau đó.

```
pop rax
ret
```

```
syscall
```

```
pop rbx
pop rdx
ret
```

```
xchg rsp,
rax
ret
```

```
add rax, 0x4
ret
```

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind[
```

```
    pop     rbp
    mov     rbp, esp
    mov     rbp, [rbp]
    mov     esp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     edx, [rax+0Ch]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx
```

```
loc_30FB:
```

```
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    mov     rax, 10h
    mov     rax, [rbp+var_28]
    mov     rax, cs:ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIcE
    lea     rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt13
    mov     rdx, cs:_ZSt4endlIcSt11char_tra
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_31BD
    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt13
    lea     rax, [rbp+var_14]
```

Return Oriented Programming (ROP)

Setup các thanh ghi gọi `execve("/bin/sh",0,0)` trên x86_64:

rax: 0x3b (**sys_execve**)

rdi: `"/bin/sh"`, **char*** pathname

rsi: 0, **char const*** argv[] (kept NULL)

rdx: 0, **char *const** envp[] (kept NULL)

syscall

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind[
mov     rbp, esp
sub     rbp, 28h
mov     [rbp+var_28], rdi
mov     [rbp+var_30], rsi
mov     rax, [rbp+var_20]
mov     eax, [rax+128h]
test    eax, eax
jnz     short loc_30FB
mov     rax, [rbp+var_28]
mov     edx, [rax+0Ch]
mov     rax, [rbp+var_28]
mov     ecx, [rax+14h]
mov     rax, [rbp+var_30]
mov     eax, [rax+10h]
mov     ebx, ecx
sub     ebx, eax
mov     eax, ebx
add     edx, eax
mov     rax, [rbp+var_28]
mov     [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
mov     rax, [rbp+var_28]
mov     eax, [rax+0Ch]
test    eax, eax
jns     loc_31C4
mov     rax, [rbp+var_28]
add     rax, 18h
mov     rsi, rax
mov     rax, cs:_ZSt4cout_ptr
mov     rdi, rax
call    __ZStlsIcSt11char_traitsIcESaIcE
lea     rsi, aIsDead ; " is dead!"
mov     rdi, rax
call    __ZStlsIcSt11char_traitsIcEERSt13
mov     rdx, cs:_ZSt4endlIcSt11char_tra
mov     rsi, rdx
mov     rdi, rax
call    __ZNSolsEPFRSoS_E ; std::ostream
mov     rax, [rbp+var_28]
mov     eax, [rax+8]
test    eax, eax
jnz     short loc_31BD
lea     rsi, aWannaCheatYes1 ; "wanna ch
mov     rax, cs:_ZSt4cout_ptr
mov     rdi, rax
call    __ZStlsIcSt11char_traitsIcEERSt13
lea     rax, [rbp+var_14]
```


Return Oriented Programming (ROP)

Trạng thái:

rax: ?

rdi: ?

rsi: ?

rdx: ?

syscall

pop rdi
pop rax
ret

pop rsi
pop rdx
ret

syscall

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
: __unwind[
```

```
    mov     rbp, esp
    sub     rbp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
```

0x401d70	mov rax, [rbp+var_28]
0x489864	mov ecx, [rax+14h]
	mov rax, [rbp+var_30]
	mov eax, [rax+10h]
	sub ebx, eax
0x3b	add ecx, eax
	mov rax, [rbp+var_28]
	mov [rax+0Ch], ecx

loc_30FB: 0x400590 : CODE KREF: Zei

	mov rax, [rbp+var_28]
	mov eax, [rax+0Ch]
	test eax, eax
0x0	jns rax, [rbp+var_28]
	add rax, 18h
0x0	mov rax, cs: _ZSt4cout_ptr
	mov rdi, rax
0x455e55	call _ZSt15IcSt11char_traitsIcESaIc

```
    call    _ZSt15IcSt11char_traitsIcESaIc
    lea     rdi, [rip+1]
    mov     rdi, rax
    call    _ZSt15IcSt11char_traitsIcESaIc
    mov     rdx, cs: _ZSt4endlIcSt11char_tra
    mov     rsi, rdx
    mov     rdi, rax
    call    _ZNSt15IcSt11char_traitsIcESaIc
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_318D
    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs: _ZSt4cout_ptr
    mov     rdi, rax
    call    _ZSt15IcSt11char_traitsIcESaIc
    lea     rax, [rbp+var_14]
```

Return Oriented Programming (ROP)

Trạng thái:

rax: ?

rdi: ?

rsi: ?

rdx: ?

syscall

rip

pop rdi
pop rax
ret

pop rsi
pop rdx
ret

syscall

```
var_30      = qword ptr -30h  
var_28      = qword ptr -28h  
var_14      = dword ptr -14h
```

```
: __unwind[
```

```
    mov     rbp, esp  
    sub     rbp, 28h  
    mov     [rbp+var_28], rdi  
    mov     [rbp+var_30], rsi
```

```
    mov     rax, [rbp+var_20]  
    mov     eax, [rax+128h]  
    test    eax, eax  
    jnz     short loc_30FB  
    mov     rax, [rbp+var_28]  
    mov     edx, [rax+0Ch]  
    mov     rax, [rbp+var_28]  
    mov     ecx, [rax+14h]
```

```
    mov     rax, [rbp+var_30]  
    mov     eax, [rax+10h]  
    sub     ebx, eax  
    add     eax, ebx  
    mov     [rbp+var_28], eax  
    mov     [rax+0Ch], edx
```

```
loc_30FB: 0x400590      CODE KREF: Zei  
    mov     rax, [rbp+var_28]  
    mov     eax, [rax+0Ch]  
    test    eax, eax  
    jns     0x0_31C4
```

```
    mov     rax, [rbp+var_28]  
    add     rax, 18h  
    mov     rax, cs:_ZSt4cout_ptr  
    mov     rdi, rax  
    call    _ZSt15IcSt11char_traitsIcESaIc@PLT
```

```
    lea     rdi, [rsi+0] ; " is dead!"  
    mov     rdi, rax  
    call    _ZSt15IcSt11char_traitsIcESaIc@PLT  
    mov     rdx, cs:_ZSt4endlIcSt11char_traitsIcESaIc@PLT  
    mov     rsi, rdx  
    mov     rdi, rax  
    call    _ZNSolsEPFRSoS_E ; std::ostream::operator<<>
```

```
    mov     rax, [rbp+var_28]  
    mov     eax, [rax+8]  
    test    eax, eax  
    jnz     short loc_318D
```

```
    lea     rsi, aWannaCheatYes1 ; "wanna cheat?"  
    mov     rax, cs:_ZSt4cout_ptr  
    mov     rdi, rax  
    call    _ZSt15IcSt11char_traitsIcESaIc@PLT  
    lea     rax, [rbp+var_14]
```

Trạng thái:

rax: ?

```
rdi: 0x489864 “/bin/sh”
```

rsi: ?

rdx: ?

syscall

rip

```
pop rdi  
pop rax  
ret
```

```
pop rsi
pop rdx
ret
```

syscall

Unwinding (ROP)

```

; __unwind {
push    rbp
rbp, esp
push    rbp
rbp, esp
push    rbp
rbp, esp
sub     esp, 28h
mov     [rbp+var_28], rdi
mov     [rbp+var_30], rsi
mov     rax, [rbp+var_28]
mov     eax, [rax+128h]
test    eax, eax
jnz     short loc_30FB
mov     rax, [rbp+var_28]
mov     edx, [rax+0Ch]
mov     rax, [rbp+var_28]
mov     ecx, [rax+14h]
mov     rax, [rbp+var_30]
mov     eax, [rax+10h]
mov     ebx, ecx
sub     ebx, eax
mov     eax, ebx
add     ecx, eax
mov     rax, [rbp+var_28]
mov     [rax+0Ch], edx
loc_30FB:
mov     rax, [rbp+var_28]
mov     eax, [rax+0Ch]
test    eax, eax
jns     loc_31C4
mov     rax, [rbp+var_28]
add     rax, 18h
mov     rax, rax
mov     rax, cs:_ZSt4cout_ptr
mov     rdi, rax
call    _ZStlsISt11char_traitsIcESaIc
lea     rsi, sDead ; " is dead!"
mov     rdi, rax
call    _ZStlsISt11char_traitsIcESaIc
mov     rdx, cs:_ZSt4endlIcSt11char_tra
mov     rsi, rdx
mov     rdi, rax
call    _ZNSoIsEPFRSoS_E ; std::ostream
mov     rax, [rbp+var_28]
mov     eax, [rax+8]
test    eax, eax
jnz     short loc_31BD
lea     rsi, aWannaCheatYes1 ; "wanna c
mov     rax, cs:_ZSt4cout_ptr
mov     rdi, rax
call    _ZStlsISt11char_traitsIcESaIc
lea     rax, [rbp+var_14]

```

Return Oriented Programming (ROP)

Trạng thái:

rax: 0x3b

rdi: 0x489864 “/bin/sh”

rsi: ?

rdx: ?

syscall

pop rdi
pop rax
ret

pop rsi
pop rdx
ret

syscall

```
var_30      = qword ptr -30h  
var_28      = qword ptr -28h  
var_14      = dword ptr -14h
```

```
; __unwind[  
    mov     rbp, esp  
    sub     esp, 28h  
    mov     [rbp+var_28], rdi  
    mov     [rbp+var_30], rsi  
    mov     rax, [rbp+var_20]  
    mov     eax, [rax+128h]  
    test    eax, eax  
    jnz     short loc_30FB  
    mov     rax, [rbp+var_28]  
    mov     edx, [rax+0Ch]  
    mov     rax, [rbp+var_28]  
    mov     ecx, [rax+14h]  
    mov     rax, [rbp+var_30]  
    mov     eax, [rax+10h]  
    mov     ebx, ecx  
    sub     ebx, eax  
    mov     eax, ebx  
    add     edx, eax  
    mov     rax, [rbp+var_28]  
    mov     [rax+0Ch], edx
```

```
loc_30FB: 0x400590      ; CODE KREF: Zei  
    mov     rax, [rbp+var_28]  
    mov     eax, [rax+0Ch]  
    test    eax, eax  
    jns     0x0_31C4  
    mov     rax, [rbp+var_28]  
    add     rax, 18h  
    mov     rax, cs:_ZSt4cout_ptr  
    mov     rdi, rax  
    call    _ZSt15IcSt11char_traitsIcESaIc  
    lea     rdi, [rip+0] ; " is dead!"  
    mov     rdi, rax  
    call    _ZSt15IcSt11char_traitsIcESaIc  
    mov     rdx, cs:_ZSt4endlIcSt11char_tra  
    mov     rsi, rdx  
    mov     rdi, rax  
    call    _ZNSolsEPFRSoS_E ; std::ostream  
    mov     rax, [rbp+var_28]  
    mov     eax, [rax+8]  
    test    eax, eax  
    jnz     short loc_318D  
    lea     rsi, aWannaCheatYes1 ; "wanna ch  
    mov     rax, cs:_ZSt4cout_ptr  
    mov     rdi, rax  
    call    _ZSt15IcSt11char_traitsIcESaIc  
    lea     rax, [rbp+var_14]
```

Return Oriented Programming (ROP)

Trạng thái:

rax: 0x3b

rdi: 0x489864 “/bin/sh”

rsi: ?

rdx: ?

syscall

```
pop rdi
pop rax
ret
```

rip

```
pop rsi
pop rdx
ret
```

```
syscall
```

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind[
push rbp
mov rbp, esp
push rbp
sub esp, 28h
mov [rbp+var_28], rdi
mov [rbp+var_30], rsi
mov rax, [rbp+var_20]
mov eax, [rax+128h]
test eax, eax
jnz short loc_30FB
mov rax, [rbp+var_28]
mov edx, [rax+0Ch]
mov rax, [rbp+var_28]
mov ecx, [rax+14h]
mov rax, [rbp+var_30]
mov eax, [rax+10h]
mov ebx, ecx
sub ebx, eax
mov eax, ebx
add edx, eax
mov rax, [rbp+var_28]
mov [rax+0Ch], edx
```

loc_30FB: ; CODE XREF: Zei

```
mov rax, [rbp+var_28]
mov eax, [rax+0Ch]
test eax, eax
jns 0x0_31C4
mov rax, [rbp+var_28]
add rax, 18h
mov rax, 0x0_rax
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call _ZSt15IcSt11char_traitsIcESaIc
lea rdi, [rsi+0] ; " is dead!"
mov rdi, rax
```

0x455e55

```
call _ZSt15IcSt11char_traitsIcESaIc
mov rdx, cs:_ZSt4endlIcSt11char_tra
mov rsi, rdx
mov rdi, rax
call _ZNSolsEPFRSoS_E ; std::ostream
mov rax, [rbp+var_28]
mov eax, [rax+8]
test eax, eax
jnz short loc_318D
lea rsi, aWannaCheatYes1 ; "wanna ch
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call _ZSt15IcSt11char_traitsIcESaIc
lea rax, [rbp+var_14]
```

Return Oriented Programming (ROP)

Trạng thái:

rax: 0x3b

rdi: 0x489864 “/bin/sh”

rsi: 0x0

rdx: ?

syscall

rip

```
pop rdi
pop rax
ret
```

```
pop rsi
pop rdx
ret
```

```
syscall
```

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr  -14h
```

```
; __unwind{
```

```
    mov     rbp, esp
    sub     rbp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     edx, [rax+0Ch]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     eax, ebx
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx
```

```
loc_30FB:
```

```
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     0x0, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    _ZStlsIcSt11char_traitsIcESaIc>::operator<< (rdi, rax)
    mov     rdi, rax
    call    _ZStlsIcSt11char_traitsIcESaIc>::operator<< (rdi, rax)
    mov     rdx, cs:_ZSt4endlIcSt11char_traitsIcESaIc>
    mov     rsi, rdx
    mov     rdi, rax
    call    _ZNSolsEPFRSoS_E (rdi, rax)
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_318D
    lea     rsi, aWannaCheatYes1 ; "wanna cheat?"
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    _ZStlsIcSt11char_traitsIcESaIc>::operator<< (rdi, rax)
    lea     rax, [rbp+var_14]
```

Return Oriented Programming (ROP)

Trạng thái:

rax: 0x3b

rdi: 0x489864 “/bin/sh”

rsi: 0x0

rdx: 0x0

syscall

rip

```
pop rdi
pop rax
ret
```

```
pop rsi
pop rdx
ret
```

```
syscall
```

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind[
push rbp
mov rbp, esp
push rbp
sub esp, 28h
mov [rbp+var_28], rdi
mov [rbp+var_30], rsi
mov rax, [rbp+var_20]
mov eax, [rax+128h]
test eax, eax
jnz short loc_30FB
mov rax, [rbp+var_28]
edx, [rax+0Ch]
mov rax, [rbp+var_28]
mov ecx, [rax+14h]
mov rax, [rbp+var_30]
mov eax, [rax+10h]
mov ebx, ecx
sub ebx, eax
mov eax, ebx
add edx, eax
mov rax, [rbp+var_28]
mov [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
mov rax, [rbp+var_28]
mov eax, [rax+0Ch]
test eax, eax
jns loc_31C4
mov rax, [rbp+var_28]
add rax, 18h
mov rsi, rax
mov rax, cs:_ZSt4cout_ptr
rdi, rax
call __ZStlsISt11char_traitsIcESaIc>
lea rdi, [rsi+0]
mov rdi, rax
call __ZStlsISt11char_traitsIcESaIc>
mov rdx, cs:_ZSt4endlIcSt11char_tra
mov rsi, rdx
mov rdi, rax
call __ZNSolsEPFRSoS_E ; std::ostrea
mov rax, [rbp+var_28]
mov eax, [rax+8]
test eax, eax
jnz short loc_318D
lea rsi, aWannaCheatYes1 ; "wanna ch
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsISt11char_traitsIcESaIc>
lea rax, [rbp+var_14]
```

0x455e55

Return Oriented Programming (ROP)

Trạng thái:

rax: 0x3b

rdi: 0x489864 “/bin/sh”

rsi: 0x0

rdx: 0x0

syscall

Shell! :D

```
pop rdi
pop rax
ret
```

```
pop rsi
pop rdx
ret
```

rip → syscall

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind[
push rbp
mov rbp, esp
push rbp
sub esp, 28h
mov [rbp+var_28], rdi
mov [rbp+var_30], rsi
mov rax, [rbp+var_20]
mov eax, [rax+128h]
test eax, eax
jnz short loc_30FB
mov rax, [rbp+var_28]
edx, [rax+0Ch]
mov rax, [rbp+var_28]
mov ecx, [rax+14h]
mov rax, [rbp+var_30]
mov eax, [rax+10h]
mov ebx, ecx
sub ebx, eax
mov eax, ebx
add edx, eax
mov rax, [rbp+var_28]
mov [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
mov rax, [rbp+var_28]
mov eax, [rax+0Ch]
test eax, eax
jns loc_31C4
mov rax, [rbp+var_28]
add rax, 18h
mov rsi, rax
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsIcSt11char_traitsIcESaIc
lea rsi, aIsDead ; " is dead!"
mov rdi, rax
call __ZStlsIcSt11char_traitsIcESaIc
mov rdx, cs:_ZSt4endlIcSt11char_tra
mov rsi, rdx
mov rdi, rax
call __ZNSolsEPFRSoS_E ; std::ostrea
mov rax, [rbp+var_28]
mov eax, [rax+8]
test eax, eax
jnz short loc_31BD
lea rsi, aWannaCheatYes1 ; "wanna ch
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsIcSt11char_traitsIcESaIc
lea rax, [rbp+var_14]
```


ROP

Công cụ tìm gadgets:

- **ROPgadget** --binary [file] (**pip install ropgadget**)
- **rp++** (<https://github.com/Overcl0k/rp>)

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h

; __unwind {
    push     rbp
    mov     rbp, rsp
    push     rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     edx, [rax+0Ch]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     eax, ebx
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx

loc_30FB:                                     ; CODE XREF: Zei
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIc@PLT
    lea     rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt13basic_stringbuf@PLT
    mov     rdx, cs:_ZSt4endlIcSt11char_traitsIc@PLT
    mov     rsi, rdx
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostream::operator<>
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_31BD

    lea     rsi, aWannaCheatYes1 ; "wanna cheat?"
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt13basic_stringbuf@PLT
    lea     rax, [rbp+var_14]
```

Better ROP

- Làm thế nào nếu như chúng ta không có đủ gadgets cần thiết (Vd: `no syscall`):
- Với những dynamically linked binaries, thư viện C chuẩn (`libc`) sẽ nằm ở đâu đó trên memory ...
- `ret2libc`: sử dụng các hàm trong libc, như `system()`

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     eax, ebx
    mov     ecx, [rbp+var_30]
    mov     [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIc
    lea     rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt13
    mov     rdx, cs:_ZSt4endlIcSt11char_tra
    mov     rsi, rdx
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_318D
    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt13
    lea     rax, [rbp+var_14]
```

Bypass ASLR

- Thông thường địa chỉ libc sẽ thay đổi trong mỗi lần chạy chương trình do cơ chế **ASLR**.
- ASLR** - Address Space Layout Randomization

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     eax, ebx
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIc
    rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt13
    mov     rdx, cs:_ZSt4endlIcSt11char_tra
    mov     rsi, rdx
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostrean
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_31BD
    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt13
    lea     rax, [rbp+var_14]
```

Bypass ASLR

????????????????

0x7f6f46f14f90

Runtime Memory

ELF Executable

Libraries (libc)

Giả sử attacker đã **leak** được địa chỉ của một hàm trong libc.

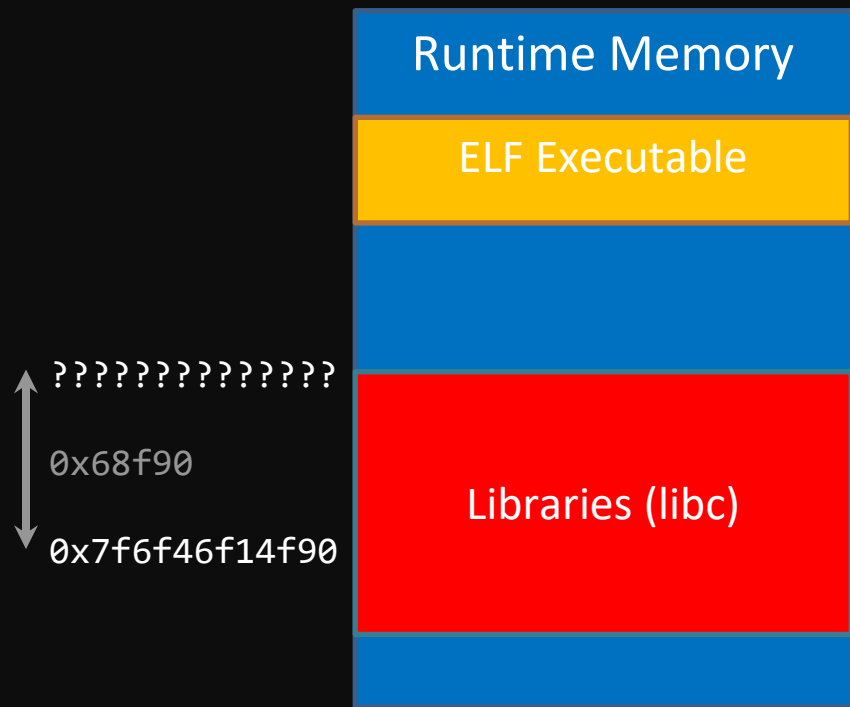
puts

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_20]
    mov     rax, [rax+14h]
    mov     ecx, [rax+14h]
    mov     ebx, ecx
    sub     ebx, eax
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIc
    lea     rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt13
    mov     rdx, cs:_ZSt4endlIcSt11char_tra
    mov     rsi, rdx
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_318D
    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt13
    lea     rax, [rbp+var_14]
```

Bypass ASLR



Giả sử attacker đã **leak** được địa chỉ của một hàm trong libc.

Với tất cả các hàm, địa chỉ libc đều **có offset so với địa chỉ base là không đổi**.

← puts

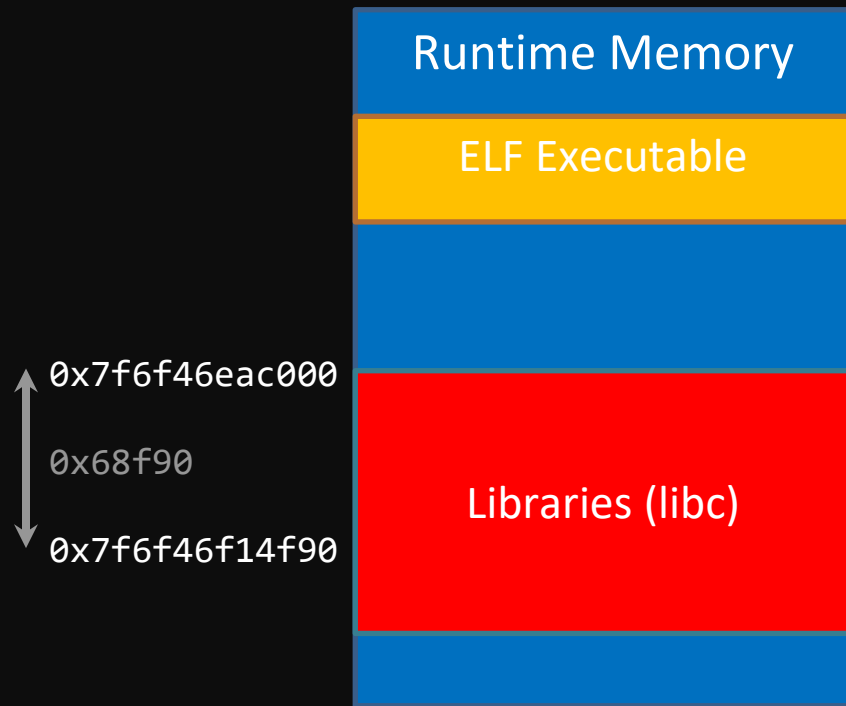
```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
```

```
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_20]
    mov     ecx, [rax+14h]
    mov     ebx, ecx
    sub     ebx, eax
    add     edx, eax
    mov     [rbp+var_28], rax
    mov     [rbp+var_14], ecx
```

```
loc_30FB:
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+8h]
    mov     eax, ecx
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIc@PLT
    lea     rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt11string@PLT
    mov     rdx, cs:_ZSt4endlIcSt11char_traitsIc@PLT
    mov     rdi, rdx
    mov     rax, cs:_ZNSolsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_31BD
    lea     rsi, aWannaCheatYes1 ; "wanna cheat?"
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt11string@PLT
    lea     rax, [rbp+var_14]
```

Bypass ASLR



Giả sử attacker đã **leak** được địa chỉ của một hàm trong libc.

Tất cả các hàm, địa chỉ libc đều **có offset so với địa chỉ base** là không đổi.

Đơn giản ta chỉ cần **cộng/trừ** là có được địa chỉ base

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr  -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
```

```
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jz      loc_30FB
    mov     ecx, [rbp+var_20]
    mov     edx, [rax+0Ch]
    mov     rax, [rbp+var_20]
    mov     rax, [rax+0Ch]
    mov     eax, [rax+0Ch]
    mov     ebx, ecx
    sub     ebx, eax
    mov     eax, ebx
    mov     edi, eax
    mov     [rbp+var_20], edi
    mov     [rbp+var_20], edi
```

```
loc_30FB:
    mov     rax, [rbp+var_20]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    mov     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    _ZStlsIcSt11char_traitsIcESaIc@PLT
    lea     rsi, aWannaCheatYes1 ; "wanna cheat?"
    mov     rax, rsi
    call    _ZStlsIcSt11char_traitsIcESaIc@PLT
    mov     rdx, cs:_ZSt4endlIcSt11char_traitsIcESaIc@PLT
    mov     rsi, rdx
    mov     rax, rsi
    call    __ZSt4endlIcSt11char_traitsIcESaIc@PLT
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_31BD
    lea     rsi, aWannaCheatYes1 ; "wanna cheat?"
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    _ZStlsIcSt11char_traitsIcESaIc@PLT
    lea     rax, [rbp+var_14]
```

Bypass ASLR

Runtime Memory

ELF Executable

Libraries (libc)

Khi có được **địa chỉ base** việc tìm địa chỉ các hàm **libc** (vd: **system**) rất đơn giản, chỉ cần **cộng offset**.

0x7f6f46eac000

0x68f90

0x7f6f46f14f90

puts

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr  -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+0Ch]
    mov     rax, [rbp+var_28]
    mov     ebx, [rax+4h]
    mov     ecx, [rax+0h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     edx, ebx
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIcE
    lea     rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt11
    mov     rdx, cs:_ZSt4endlIcSt11char_tra
    mov     rdi, rax
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_31BD
    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcEERSt11
    lea     rax, [rbp+var_14]
```

Bypass ASLR

Địa chỉ
puts

libc
base

offset
của
system

```
pwndbg> p puts
$3 = {int (const char *)} 0x7ffff7e06ed0 <__GI_IO_puts>
pwndbg> vmmmap libc
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
      Start      End Perm      Size Offset File
0x7ffff7d83000 0x7ffff7d86000 rw-p      3000      0 [anon_7ffff7d83]
0x7ffff7d86000 0x7ffff7dae000 r--p     28000      0 /usr/lib/x86_64-linux-gnu/libc.so.6
0x7ffff7dae000 0x7ffff7f43000 r-xp    195000  28000 /usr/lib/x86_64-linux-gnu/libc.so.6
0x7ffff7f43000 0x7ffff7f9b000 r--p     58000  1bd000 /usr/lib/x86_64-linux-gnu/libc.so.6
0x7ffff7f9b000 0x7ffff7f9f000 r--p      4000  214000 /usr/lib/x86_64-linux-gnu/libc.so.6
0x7ffff7f9f000 0x7ffff7fa1000 rw-p      2000  218000 /usr/lib/x86_64-linux-gnu/libc.so.6
0x7ffff7dae000 0x7ffff7f43000 r-xp    195000  28000 /usr/lib/x86_64-linux-gnu/libc.so.6
pwndbg> p system
$4 = {int (const char *)} 0x7ffff7dd6d60 <__libc_system>
pwndbg> p/x 0x7ffff7dd6d60-0x7ffff7d86000
$5 = 0x50d60
pwndbg> A|
```

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_20]
    mov     eax, [rax+128h]
```

To boldly go where no
shell has gone before

```
mov     rsi, rdx
mov     rdi, rax
call    __ZNSolsEPFRSoS_E ; std::ostream
mov     rax, [rbp+var_28]
mov     eax, [rax+8]
test    eax, eax
jnz     short loc_318D
lea     rsi, aWannaCheatYes1 ; 'wanna ch
mov     rax, cs:ZSt4cout_ptr
mov     rdi, rax
call    __ZStlsISt11char_traitsIcEERSt13
mov     rax, [rbp+var_14]
```


Even better ROP

Làm thế nào để leak địa chỉ?

```
var_30      = qword ptr -30h
var_28      = qword ptr -28h
var_14      = dword ptr -14h
```

```
; __unwind {
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
```

```
    mov     rax, [rbp+var_28]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     edx, [rax+0Ch]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    sub     ebx, eax
    mov     eax, ebx
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx
```

```
loc_30FB:                                     ; CODE XREF: Zei
    mov     rax, [rbp+var_28]
    mov     eax, [rax+0Ch]
    test    eax, eax
    jns     loc_31C4
    mov     rax, [rbp+var_28]
    add     rax, 18h
    mov     rsi, rax
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsIcSt11char_traitsIcESaIc
    lea     rsi, aIsDead ; " is dead!"
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt13
    mov     rdx, cs:_ZSt4endlIcSt11char_tra
    mov     rsi, rdx
    mov     rdi, rax
    call    __ZNSolsEPFRSoS_E ; std::ostream
    mov     rax, [rbp+var_28]
    mov     eax, [rax+8]
    test    eax, eax
    jnz     short loc_318D
```

```
    lea     rsi, aWannaCheatYes1 ; "wanna ch
    mov     rax, cs:_ZSt4cout_ptr
    mov     rdi, rax
    call    __ZStlsISt11char_traitsIcEERSt13
    lea     rax, [rbp+var_14]
```

```
; __unwind{
    push    rbp
    mov     rbp, rsp
    push    rbx
    sub     rsp, 28h
    mov     [rbp+var_28], rdi
    mov     [rbp+var_30], rsi
    mov     rax, [rbp+var_28]
    mov     eax, [rax+128h]
    test    eax, eax
    jnz     short loc_30FB
    mov     rax, [rbp+var_28]
    mov     edx, [rax+0Ch]
    mov     rax, [rbp+var_28]
    mov     ecx, [rax+14h]
    mov     rax, [rbp+var_30]
    mov     eax, [rax+10h]
    mov     ebx, ecx
    mov     sub     ebx, eax
    mov     eax, ebx
    add     edx, eax
    mov     rax, [rbp+var_28]
    mov     [rax+0Ch], edx
}
```

- ```
mov rax, [rbp+var_28]
mov eax, [rax+0Ch]
test eax, eax
jns loc_31C4
mov rax, [rbp+var_28]
add rsp, 0xd8)
mov rsi, 0
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsISt11char_traitsIcESaIc>E9writeIsc&aIsDead ; " is dead!"
mov rdi, rax
call __ZStlsISt11char_traitsIcEEERSt13_Iterator>E9writeIsc&aWannaCheatYes1 ; "wanna cheat?"
mov rdx, cs:_ZSt4endlIcSt11char_traitsIc>E9writeIsc&aWannaCheatYes1 ; std::ostream<char>::operator<<(const char*) const
mov rsi, rdx
mov rdi, rax
call __ZNSoLseEPFRSoS_E ; std::ostream<char>::operator<<(const char*) const
mov rax, [rbp+var_28]
mov eax, [rax+8]
test eax, eax
jnz short loc_31BD
lea rsi, aWannaCheatYes1 ; "wanna cheat?"
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsISt11char_traitsIcEEERSt13_Iterator>E9writeIsc&aWannaCheatYes1 ; std::ostream<char>::operator<<(const char*) const
lea rax, [rbp+var_14]
```

```
sub rax, 0xd8)
```

```
push r12
mov rdi, rax
call @ZStlsIcSt11char_3L1E@
lea rsi, aIsDead
```

# Dynamic Linking

## GOT - Global Offset Table

- Mảng function table

## PLT - Procedure Linkage Table

- Gọi `*got[function_offset]()`

```
var_30 = qword ptr -30h
var_28 = qword ptr -28h
var_14 = dword ptr -14h
```

```
; __unwind {
 push rbp
 mov rbp, rsp
 push rbx
 sub rsp, 28h
 mov [rbp+var_28], rdi
 mov [rbp+var_30], rsi
 mov rax, [rbp+var_20]
 mov eax, [rax+128h]
 test eax, eax
 jnz short loc_30FB
 mov rax, [rbp+var_28]
 mov edx, [rax+0Ch]
 mov rax, [rbp+var_28]
 mov ecx, [rax+14h]
 mov rax, [rbp+var_30]
 mov eax, [rax+10h]
 mov ebx, ecx
 sub ebx, eax
 mov eax, ebx
 add edx, eax
 mov rax, [rbp+var_28]
 mov [rax+0Ch], edx
}
```

```
loc_30FB: ; CODE XREF: Zei
 mov rax, [rbp+var_28]
```

```
0x400480 <gets@plt+0>: jmp QWORD PTR [rip+0x200b9a] # 0x601020
```

---GOT---

```
0x601018 --> 0x7f21c0e88190 (<printf>: sub rsp, 0xd8)
```

```
0x601020 --> 0x7f21c0ea1660 (<gets>: push r12)
```

```
mov rax, [rbp+var_28]
mov eax, [rax+8]
test eax, eax
jnz short loc_318D
```

```
lea rsi, aWannaCheatYes1 ; "wanna ch
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsISt11char_traitsIcEERSt13
lea rax, [rbp+var_14]
```

# Even better ROP

Liệu chúng ta có thể tận dụng **GOT** và **PLT** để leak libc ?

Hint: **puts(puts@got); main();**

```
var_30 = qword ptr -30h
var_28 = qword ptr -28h
var_14 = dword ptr -14h
```

```
; __unwind {
 push rbp
 mov rbp, rsp
 push rbx
 sub rsp, 28h
 mov [rbp+var_28], rdi
 mov [rbp+var_30], rsi
 mov rax, [rbp+var_20]
 mov eax, [rax+128h]
 test eax, eax
 jnz short loc_30FB
 mov rax, [rbp+var_28]
 mov edx, [rax+0Ch]
 mov rax, [rbp+var_28]
 mov ecx, [rax+14h]
 mov rax, [rbp+var_30]
 mov eax, [rax+10h]
 mov ebx, ecx
 sub ebx, eax
 mov eax, ebx
 add edx, eax
 mov rax, [rbp+var_28]
 mov [rax+0Ch], edx
}
```

```
loc_30FB: ; CODE XREF: Zei
 mov rax, [rbp+var_28]
 mov eax, [rax+0Ch]
 test eax, eax
 jns loc_31C4
 mov rax, [rbp+var_28]
 add rax, 18h
 mov rsi, rax
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsIcSt11char_traitsIcESaIc
 lea rsi, aIsDead ; " is dead!"
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt13
 mov rdx, cs:_ZSt4endlIcSt11char_tra
 mov rsi, rdx
 mov rdi, rax
 call __ZNSolsEPFRSoS_E ; std::ostream
 mov rax, [rbp+var_28]
 mov eax, [rax+8]
 test eax, eax
 jnz short loc_318D
 lea rsi, aWannaCheatYes1 ; "wanna ch
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt13
 lea rax, [rbp+var_14]
```

# ROPchain is so damm big

- Nếu như ROPchain của chúng ta quá lớn nhưng stack lại không đủ để chứa? :

1. Ghi đè `$rsp` đi nơi khác (Fake stack ?)
2. Disable DEP và thực thi shellcode ?

```
var_30 = qword ptr -30h
var_28 = qword ptr -28h
var_14 = dword ptr -14h

; unwind f
push rbp
mov rbp, rsp
push rbx
sub rsp, 28h
mov [rbp+var_28], rdi
mov [rbp+var_30], rsi
mov rax, [rbp+var_20]
mov eax, [rax+128h]
test eax, eax
jnz short loc_30FB
mov rax, [rbp+var_28]
mov ecx, [rax+14h]
mov rax, [rbp+var_30]
mov eax, [rax+10h]
mov ebx, ecx
sub ebx, eax
mov edx, ebx
add edx, eax
mov rax, [rbp+var_28]
mov [rax+0Ch], edx

loc_30FB:
; CODE XREF: Zei
mov rax, [rbp+var_28]
mov eax, [rax+0Ch]
test eax, eax
jns loc_31C4
mov rax, [rbp+var_28]
add rax, 18h
mov rsi, rax
mov rdi, rax
call __ZStlsIcSt11char_traitsIcESaIc
lea rsi, aIsDead ; " is dead!"
mov rdi, rax
call __ZStlsIcSt11char_traitsIcEERSt13
mov rdx, cs:_ZSt4endlIcSt11char_tra
mov rsi, rdx
mov rdi, rax
call __ZNSolsEPFRSoS_E ; std::ostream
mov rax, [rbp+var_28]
mov eax, [rax+8]
test eax, eax
jnz short loc_31BD
lea rsi, aWannaCheatYes1 ; "wanna ch
mov rax, cs:_ZSt4cout_ptr
mov rdi, rax
call __ZStlsIcSt11char_traitsIcEERSt13
lea rax, [rbp+var_14]
```

# Stack pivot

- Chúng ta có thể ghi đè **\$rsp** (stack pointer) đi nơi khác, khi đó chương trình sẽ hiểu địa chỉ **\$rsp** trở tới là stack.
- Gadget hữu ích:
  - pop rsp** : rất hiếm khi thấy. Nhưng nếu thấy, thì bạn là một người rất may mắn 😊
  - xchg <reg>, rsp** : hoán đổi giá trị của <reg> và rsp với nhau
  - leave** : hoạt động như **mov rsp, rbp ; pop rbp**

```
var_30 = qword ptr -30h
var_28 = qword ptr -28h
var_14 = dword ptr -14h

; __unwind {
 push rbp
 mov rbp, rsp
 push rbx
 sub rsp, 28h
 mov [rbp+var_28], rdi
 mov [rbp+var_30], rsi
 mov rax, [rbp+var_20]
 mov eax, [rax+128h]
 test eax, eax
 jnz short loc_30FB
 mov rax, [rbp+var_20]
 mov rax, [rbp+var_28]
 mov ecx, [rax+14h]
 mov rax, [rbp+var_30]
 mov eax, [rax+10h]
 mov ebx, ecx
 sub ebx, eax
 mov eax, ebx
 add edx, eax
 mov rax, [rbp+var_28]
 mov [rax+0Ch], edx

loc_30FB:
 mov eax, [rax+0Ch]
 test eax, eax
 jns loc_31C4
 mov rax, [rbp+var_28]
 add rax, 18h
 mov rsi, rax
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsIcSt11char_traitsIcESaIc>@8
 mov rsi, aIsDead ; " is dead!"
 mov rdi, rax
 call __ZStlsIcSt11char_traitsIcESaIc>@8
 mov rdx, cs:_ZSt4endlIcSt11char_traitsIc>@8
 mov rsi, rdx
 mov rdi, rax
 call __ZNSoIsEPFRSoS_E ; std::ostream::operator<>
 mov rax, [rbp+var_28]
 mov eax, [rax+8]
 test eax, eax
 jnz short loc_31BD

 lea rsi, aWannaCheatYes1 ; "wanna cheat?"
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsIcSt11char_traitsIcESaIc>@8
 lea rax, [rbp+var_14]
```

# Ret 2 mprotect

- Linux có hàm **mprotect** giúp ta thay đổi quyền của một vùng bộ nhớ.
- Ta có thể thay đổi quyền của các data segment như **stack/heap/bss** có quyền **write** lẫn **execute**.

```
var_30 = qword ptr -30h
var_28 = qword ptr -28h
var_14 = dword ptr -14h

; __unwind {
 push rbp
 mov rbp, rsp
 push rbx
 sub rsp, 28h
 mov [rbp+var_28], rdi
 mov [rbp+var_30], rsi
 mov rax, [rbp+var_20]
 mov eax, [rax+128h]
 test eax, eax
 jnz short loc_30FB
 mov rax, [rbp+var_28]
 mov ecx, [rax+14h]
 mov rax, [rbp+var_30]
 mov ebx, ecx
 sub ebx, eax
 mov eax, ebx
 add edx, eax
 mov rax, [rbp+var_28]
 mov [rax+0Ch], edx

loc_30FB: ; CODE XREF: Zei
 mov rax, [rbp+var_28]
 mov eax, [rax+0Ch]
 test eax, eax
 jns loc_31C4
 mov rax, [rbp+var_28]
 add rax, 18h
 mov rsi, rax
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsIcSt11char_traitsIcESaIc
 lea rsi, aIsDead ; " is dead!"
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt13
 mov rdx, cs:_ZSt4endlIcSt11char_tra
 mov rsi, rdx
 mov rdi, rax
 call __ZNSolsEPFRSoS_E ; std::ostream
 mov rax, [rbp+var_28]
 mov eax, [rax+8]
 test eax, eax
 jnz short loc_31BD
 lea rsi, aWannaCheatYes1 ; "wanna ch
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt13
 lea rax, [rbp+var_14]
```

# Ret 2 mprotect

- Đơn giản chỉ cần gọi **mprotect(address,size,7)**  
(<https://man7.org/linux/man-pages/man2/mprotect.2.html>)
- Kĩ thuật **ret2shellcode** giờ có thể áp dụng.

```
var_30 = qword ptr -30h
var_28 = qword ptr -28h
var_14 = dword ptr -14h

; __unwind {
 push rbp
 mov rbp, rsp
 push rbx
 sub rsp, 28h
 mov [rbp+var_28], rdi
 mov [rbp+var_30], rsi
 mov rax, [rbp+var_20]
 mov eax, [rax+128h]
 test eax, eax
 jnz short loc_30FB
 mov rax, [rbp+var_28]
 mov edx, [rax+0Ch]
 mov rax, [rbp+var_28]
 mov ecx, [rax+14h]
 mov rax, [rbp+var_30]
 mov eax, [rax+10h]
 mov ebx, ecx
 sub ebx, eax
 mov eax, ebx
 add edx, eax
 mov rax, [rbp+var_28]
 mov [rax+0Ch], edx

loc_30FB: ; CODE XREF: Zei
 mov rax, [rbp+var_28]
 mov eax, [rax+0Ch]
 test eax, eax
 jns loc_31C4
 mov rax, [rbp+var_28]
 add rax, 18h
 mov rsi, rax
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsIcSt11char_traitsIcESaIc
 lea rsi, aIsDead ; " is dead!"
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt13
 mov rdx, cs:_ZSt4endlIcSt11char_tra
 mov rsi, rdx
 mov rdi, rax
 call __ZNSolsEPFRSoS_E ; std::ostream
 mov rax, [rbp+var_28]
 mov eax, [rax+8]
 test eax, eax
 jnz short loc_31BD
 lea rsi, aWannaCheatYes1 ; "wanna ch
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt13
 lea rax, [rbp+var_14]
```



# Practice

## Wargames:

- [microcorruption.com](https://microcorruption.com)
- [pwnable.kr](https://pwnable.kr)
- [pwnable.tw](https://pwnable.tw)
- [ropemporium.com](https://ropemporium.com)

```
var_30 = qword ptr -30h
var_28 = qword ptr -28h
var_14 = dword ptr -14h
```

```
; __unwind {
 push rbp
 mov rbp, rsp
 push rbx
 sub rsp, 28h
 mov [rbp+var_28], rdi
 mov [rbp+var_30], rsi
 mov rax, [rbp+var_20]
 mov eax, [rax+128h]
 test eax, eax
 jnz short loc_30FB
 mov rax, [rbp+var_28]
 mov edx, [rax+0Ch]
 mov rax, [rbp+var_28]
 mov ecx, [rax+14h]
 mov rax, [rbp+var_30]
 mov eax, [rax+10h]
 mov ebx, ecx
 sub ebx, eax
 mov eax, ebx
 add edx, eax
 mov rax, [rbp+var_28]
 mov [rax+0Ch], edx
```

```
loc_30FB: ; CODE XREF: Zei
 mov rax, [rbp+var_28]
 mov eax, [rax+0Ch]
 test eax, eax
 jns loc_31C4
 mov rax, [rbp+var_28]
 add rax, 18h
 mov rsi, rax
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsIcSt11char_traitsIcESaIc
 lea rsi, aIsDead ; " is dead!"
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt11
 mov rdx, cs:_ZSt4endlIcSt11char_tra
 mov rdi, rdx
 mov rsi, rdx
 mov rdi, rax
 call __ZNSolsEPFRSoS_E ; std::ostream
 mov rax, [rbp+var_28]
 mov eax, [rax+8]
 test eax, eax
 jnz short loc_31BD
 lea rsi, aWannaCheatYes1 ; "wanna ch
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt11
 lea rax, [rbp+var_14]
```

# Reference

- <https://github.com/RPISEC/MBE>
- <https://ir0nstone.gitbook.io/notes/types/stack/stack-pivoting>

```
var_30 = qword ptr -30h
var_28 = qword ptr -28h
var_14 = dword ptr -14h

; __unwind {
 push rbp
 mov rbp, rsp
 push rbx
 sub rsp, 28h
 mov [rbp+var_28], rdi
 mov [rbp+var_30], rsi
 mov rax, [rbp+var_20]
 mov eax, [rax+128h]
 test eax, eax
 jnz short loc_30FB
 mov rax, [rbp+var_28]
 mov edx, [rax+0Ch]
 mov rax, [rbp+var_28]
 mov ecx, [rax+14h]
 mov rax, [rbp+var_30]
 mov rax, [rax+10h]
 mov ebx, ecx
 sub ebx, eax
 mov eax, ebx
 add edx, eax
 mov rax, [rbp+var_28]
 mov [rax+0Ch], edx

loc_30FB: ; CODE XREF: Zei
 mov rax, [rbp+var_28]
 mov eax, [rax+0Ch]
 test eax, eax
 jns loc_31C4
 mov rax, [rbp+var_28]
 add rax, 18h
 mov rsi, rax
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsIcSt11char_traitsIcESaIc
 lea rsi, aIsDead ; " is dead!"
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt13
 mov rdx, cs:_ZSt4endlIcSt11char_tra
 mov rsi, rdx
 mov rdi, rax
 call __ZNSolsEPFRSoS_E ; std::ostream
 mov rax, [rbp+var_28]
 mov eax, [rax+8]
 test eax, eax
 jnz short loc_31BD

 lea rsi, aWannaCheatYes1 ; "wanna ch
 mov rax, cs:_ZSt4cout_ptr
 mov rdi, rax
 call __ZStlsISt11char_traitsIcEERSt13
 lea rax, [rbp+var_14]
```