2025.10.12

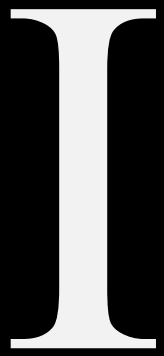
Poisoning in 2.35

가디언 시스템 보안 & 취약점 분석 세미나 5.1

임준서

Recap

Use After Free, Double Free



Use After free

Concept

Free 한 메모리의 재사용을 의미

보통 dynamic memory는 재사용 된다

free 된 후 use → 다시 malloc → 조작된 chunk를 해석

Realloc으로도 동일한 기능을 얻을 수 있다.

Use After Free

Concept

사용자가 해당 포인터를 조작

→ chunk의 metadata 조작

→ 원하는 위치에 malloc

그 위치가 GOT라면?

Double Free

Concept

2번 free하는 것을 의미

2번 free하면 bin에 있으면서도 allocated

tcachebin [A] 〉 [A] 〉 null

이후 allocated [A] , tcachebin 〉 [A] 〉 null

Tcachebin Poisoning

glibc 2.27

UaF를 통해 next 포인터를 조작

→ 다음에 allocate 될 위치를 지정

해당 위치로 malloc 받고 값을 적는다.

Tcachebin Poisoning

glibc 2.27

UaF는 어떻게 획득?

개발자 실수

보통 boundary에서 논리 오류 발생

Tcachebin Poisoning

Glibc 2.35

II

Security Mitigations 1

Safe Linking

next 포인터에 대해 SafeLink

```
next ⊕ (heap_addr >> 12)
```

즉, heap 주소에 대한 leak가 필요하다.

Safe Linking

Example

tcachebin

0x20	0x0
0x30	0x0
…	…

0xA0	user_data	
0x90		
0x80	mchunk_prev_size	0x30
0x70	user_data	
0x60	mchunk_prev_size	0x20
0x50	user_data	
0x40		
0x30	mchunk_prev_size	0x30
0x20	user_data	
0x10		
0x00	mchunk_prev_size	0x30

heap_base

Safe Linking

Example

tcachebin

0x20	0x0
0x30	0x0
…	…

0xA0	user_data	
0x90		
0x80	mchunk_prev_size	0x30
0x70	user_data	
0x60	mchunk_prev_size	0x20
0x50	user_data	
0x40		
0x30	mchunk_prev_size	0x30
0x20	user_data	
0x10		
0x00	mchunk_prev_size	0x30

heap_base

Safe Linking

Example

```
free(p1);
```

tcachebin

0x20	0x0
0x30	heap_base+0x10
…	…

0xA0	user_data	
0x90		
0x80	mchunk_prev_size	0x30
0x70	user_data	
0x60	mchunk_prev_size	0x20
0x50	user_data	
0x40		
0x30	mchunk_prev_size	0x30
0x20		
0x10	heap_base>>12	–
0x00	mchunk_prev_size	0x31

heap_base

Safe Linking

Source

```
1  #define PROTECT_PTR(pos, ptr) \
2    ((__typeof (ptr)) ((((size_t) pos) >> 12) ^ ((size_t) ptr)))
3  #define REVEAL_PTR(ptr)  PROTECT_PTR (&ptr, ptr)
```

Macro

elixir.bootlin.com/glibc/glibc-2.35/source/malloc/malloc.c

임준서

Security Mitigations 2

Key in tcache

```
1  if (__glibc_unlikely (e->key == tcache_key))
2    {
3      tcache_entry *tmp;
4      size_t cnt = 0;
5      LIBC_PROBE (memory_tcache_double_free, 2, e, tc_idx);
6      for (tmp = tcache->entries[tc_idx];
7       tmp;
8       tmp = REVEAL_PTR (tmp->next), ++cnt)
9          {
10     if (cnt >= mp_.tcache_count)
11       malloc_printerr ("free(): too many chunks detected in tcache");
12     if (__glibc_unlikely (!aligned_OK (tmp)))
13       malloc_printerr ("free(): unaligned chunk detected in tcache 2");
14     if (tmp == e)
15       malloc_printerr ("free(): double free detected in tcache 2");
16     /* If we get here, it was a coincidence.  We've wasted a
17        few cycles, but don't abort.  */
18         }
19   }
```

Free 과정 일부

elixir.bootlin.com/glibc/glibc-2.35/source/malloc/malloc.c

임준서

Key in tcache

Concept

fd 포인터 다음에 key 값 추가

"key값 == 지정된 값"은 곧 double free

이때 "지정된 값"은 상수

free시 생성된 key를 지우기 위해서 재할당시 malloc+0x8의 메모리는 0으로 지워진다.

임준서

Security Mitigations 3

Counter in tcache

```
1  if (tc_idx < mp_.tcache_bins
2      && tcache
3      && tcache->counts[tc_idx] > 0)
4    {
5      victim = tcache_get (tc_idx);
6      return tag_new_usable (victim);
7    }
```

Malloc

elixir.bootlin.com/glibc/glibc-2.35/source/malloc/malloc.c

임준서

Counter in tcache

Concept

cnt 값이 0이면 tcache 건너뜀

free시 생성된 key를 지우기 위해서 재할당시 malloc+0x8의 메모리는 0으로 지워진다.

임준서

Bypass

Safe Linking

첫 번째 malloc은 safelink(null)

malloc 〉 free 〉 malloc 으로 leak

Bypass

Key in tcache

key값은 상수이므로 아무 수나 대입하면 통과

→ 0x10 바이트 오버라이트

Bypass

Counter in tcache

Double Free하더라도 last chunk는 반영 x

count에 유의해서 가짜 청크를 생성

Exploit

Glibc 2.35

III

실습 / HW [kidheap]

dreamhack.io/wargame/challenges/1887

세미나에서는 AAR/AAW 획득까지 다룸

지금까지 배운 내용으로 풀린다...!

Unsorted bin leak

임준서

2025.10.12

Q&A

질문이 있다면 하십시오

임준서

2.5cm-2.5cm 떨어진 제목 36px

제목 하단의 부제목 18px

3.5cm 떨어진 내용 1 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

3.5cm 떨어진 내용 2 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

3.5cm 떨어진 내용 3 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

1cm-1cm 떨어진 주석 12px

1cm-1cm 떨어진 주석 12px

임준서

2.5cm-3.5cm 떨어진 제목 36px

제목 하단의 부제목 18px

3.5cm 떨어진 내용 1 32px

```
Git init
Git status
Git add text.txt
Git add .
Git commit
Ctrl+C
Git commit -m “genesis”
Git log
Git log --oneline
```

```
Git add .
Git reset .
Git commit -m “add README”
Git log --oneline -n 3
Git commit –a –m “hello”
```

1cm-1cm 떨어진 주석 12px

1cm-1cm 떨어진 주석 12px

중심에서 0.3cm 떨어진 소속 18px