309552017 黃子庭 lab7

https://hackmd.io/@FNtGY_zjTEKLa0UbkzDOXw/309552017_lab7
(https://hackmd.io/@FNtGY_zjTEKLa0UbkzDOXw/309552017_lab7)
編譯器：gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0

1. 下面是常見的漏洞，請分別寫出有下面漏洞的簡單程式，並告訴我 Valgrind 和 ASan 兩個分別找不找的出來

- **Heap out-of-bounds read/write**

```c
// 有問題的程式碼
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define BUFSIZE 8

int main(int argc, char **argv) {
  char *buf;
  buf = (char *)malloc(sizeof(char)*BUFSIZE);
  strcpy(buf, argv[1]);

  return 0;
}

```

ASan report

```
estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ ./H 123456789
=================================================================
==10010==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000018 at pc
WRITE of size 10 at 0x602000000018 thread T0
    #0 0x7fda92c7c3a5  (/usr/lib/x86_64-linux-gnu/libasan.so.4+0x663a5)
    #1 0x55fcbe0f08e0 in main (/home/estee/workspace/SoftwareTest/lab7/H+0x8e0)
    #2 0x7fda92846bf6 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21bf6)
    #3 0x55fcbe0f07a9 in _start (/home/estee/workspace/SoftwareTest/lab7/H+0x7a9)

0x602000000018 is located 0 bytes to the right of 8-byte region [0x602000000010,0x6020
allocated by thread T0 here:
    #0 0x7fda92cf4b40 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+
    #1 0x55fcbe0f08a2 in main (/home/estee/workspace/SoftwareTest/lab7/H+0x8a2)
    #2 0x7fda92846bf6 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21bf6)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/usr/lib/x86_64-linux-gnu/libasan.so.
Shadow bytes around the buggy address:
  0x0c047fff7fb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c047fff8000: fa fa 00[fa]fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8010: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==10010==ABORTING
```

```
valgrind report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ valgrind ./a.out 123456789
==10008== Memcheck, a memory error detector
==10008== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10008== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10008== Command: ./a.out 123456789
==10008== 
==10008== Invalid write of size 1
==10008==    at 0x4C34E00: strcpy (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux
==10008==    by 0x1086C0: main (in /home/estee/workspace/SoftwareTest/lab7/a.out)
==10008==  Address 0x522f048 is 0 bytes after a block of size 8 alloc'd
==10008==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux
==10008==    by 0x1086A2: main (in /home/estee/workspace/SoftwareTest/lab7/a.out)
==10008== 
==10008== Invalid write of size 1
==10008==    at 0x4C34E0D: strcpy (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux
==10008==    by 0x1086C0: main (in /home/estee/workspace/SoftwareTest/lab7/a.out)
==10008==  Address 0x522f049 is 1 bytes after a block of size 8 alloc'd
==10008==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux
==10008==    by 0x1086A2: main (in /home/estee/workspace/SoftwareTest/lab7/a.out)
==10008== 
==10008== 
==10008== HEAP SUMMARY:
==10008==     in use at exit: 8 bytes in 1 blocks
==10008==   total heap usage: 1 allocs, 0 frees, 8 bytes allocated
==10008== 
==10008== LEAK SUMMARY:
==10008==    definitely lost: 8 bytes in 1 blocks
==10008==    indirectly lost: 0 bytes in 0 blocks
==10008==      possibly lost: 0 bytes in 0 blocks
==10008==    still reachable: 0 bytes in 0 blocks
==10008==         suppressed: 0 bytes in 0 blocks
==10008== Rerun with --leak-check=full to see details of leaked memory
==10008== 
==10008== For counts of detected and suppressed errors, rerun with: -v
==10008== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```

ASan 能 , valgrind 能

- **Stack out-of-bounds read/write**

```c
//有問題的程式碼
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main()
{
  int a[8] = {0};
  a[8] = 100;

  return 0;
}
```

ASan report

```
estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ ./S
=================================================================
==10045==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffe049747f0 at p
WRITE of size 4 at 0x7ffe049747f0 thread T0
    #0 0x55d816956abf in main (/home/estee/workspace/SoftwareTest/lab7/S+0xabf)
    #1 0x7f6d9197dbf6 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21bf6)
    #2 0x55d816956899 in _start (/home/estee/workspace/SoftwareTest/lab7/S+0x899)

Address 0x7ffe049747f0 is located in stack of thread T0 at offset 64 in frame
    #0 0x55d816956989 in main (/home/estee/workspace/SoftwareTest/lab7/S+0x989)

  This frame has 1 object(s):
    [32, 64) 'a' <== Memory access at offset 64 overflows this variable
HINT: this may be a false positive if your program uses some custom stack unwind mecha
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow (/home/estee/workspace/SoftwareTest/l
Shadow bytes around the buggy address:
  0x1000409268a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000409268b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000409268c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000409268d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000409268e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x1000409268f0: 00 00 00 00 00 00 f1 f1 f1 f1 00 00 00 00[f3]f3
  0x100040926900: f3 f3 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100040926910: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100040926920: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100040926930: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100040926940: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==10045==ABORTING
```

```
valgrind report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ valgrind ./a.out
==10046== Memcheck, a memory error detector
==10046== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10046== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10046== Command: ./a.out
==10046==
==10046==
==10046== HEAP SUMMARY:
==10046==     in use at exit: 0 bytes in 0 blocks
==10046==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==10046==
==10046== All heap blocks were freed -- no leaks are possible
==10046==
==10046== For counts of detected and suppressed errors, rerun with: -v
==10046== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

ASan 能 , valgrind 不能

Global out-of-bounds read/write

```
// 有問題的程式碼
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int a[8] = {0};

int main()
{
  a[8] = 100;

  return 0;
}
```

```
ASan report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ ./G
=================================================================
==10108==ERROR: AddressSanitizer: global-buffer-overflow on address 0x555c8062d0c0 at
WRITE of size 4 at 0x555c8062d0c0 thread T0
    #0 0x555c8042c98f in main (/home/estee/workspace/SoftwareTest/lab7/G+0x98f)
    #1 0x7f7387506bf6 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21bf6)
    #2 0x555c8042c879 in _start (/home/estee/workspace/SoftwareTest/lab7/G+0x879)

0x555c8062d0c0 is located 0 bytes to the right of global variable 'a' defined in 'G.c:
SUMMARY: AddressSanitizer: global-buffer-overflow (/home/estee/workspace/SoftwareTest/
Shadow bytes around the buggy address:
  0x0aac100bd9c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aac100bd9d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aac100bd9e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aac100bd9f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aac100bda00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0aac100bda10: 00 00 00 00 00 00 00 00[f9]f9 f9 f9 00 00 00 00
  0x0aac100bda20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aac100bda30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aac100bda40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aac100bda50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0aac100bda60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==10108==ABORTING
```

```
valgrind report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ valgrind ./a.out
==10109== Memcheck, a memory error detector
==10109== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10109== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10109== Command: ./a.out
==10109==
==10109==
==10109== HEAP SUMMARY:
==10109==     in use at exit: 0 bytes in 0 blocks
==10109==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==10109==
==10109== All heap blocks were freed -- no leaks are possible
==10109==
==10109== For counts of detected and suppressed errors, rerun with: -v
==10109== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

ASan 能 , valgrind 不能

Use-after-free

```c
// 有問題的程式碼
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main()
{
  char *str = malloc(4);
  free(str);

  printf("%s\n", str);

  return 0;
}
```

```
ASan report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ ./UF
```

```
valgrind report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ valgrind ./a.out
==10139== Memcheck, a memory error detector
==10139== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10139== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10139== Command: ./a.out
==10139==
==10139== Invalid read of size 1
==10139==    at 0x4C34CF2: strlen (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux
==10139==    by 0x4EBEAB1: puts (ioputs.c:35)
==10139==    by 0x1086F7: main (in /home/estee/workspace/SoftwareTest/lab7/a.out)
==10139==  Address 0x522f040 is 0 bytes inside a block of size 4 free'd
==10139==    at 0x4C32D3B: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.s
==10139==    by 0x1086EB: main (in /home/estee/workspace/SoftwareTest/lab7/a.out)
==10139==  Block was alloc'd at
==10139==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux
==10139==    by 0x1086DB: main (in /home/estee/workspace/SoftwareTest/lab7/a.out)
==10139==

==10139==
==10139== HEAP SUMMARY:
==10139==     in use at exit: 0 bytes in 0 blocks
==10139==   total heap usage: 2 allocs, 2 frees, 1,028 bytes allocated
==10139==
==10139== All heap blocks were freed -- no leaks are possible
==10139==
==10139== For counts of detected and suppressed errors, rerun with: -v
==10139== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

ASan 不能 , valgrind 能

- **Use-after-return**

```c
// 有問題的程式碼
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main()
{
  int a = 100;

  return 0;

  printf("%d\n", a);
}
```

```
ASan report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ ./UR
```

```
valgrind report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ valgrind ./a.out
==10173== Memcheck, a memory error detector
==10173== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10173== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==10173== Command: ./a.out
==10173==
==10173==
==10173== HEAP SUMMARY:
==10173==     in use at exit: 0 bytes in 0 blocks
==10173==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==10173==
==10173== All heap blocks were freed -- no leaks are possible
==10173==
==10173== For counts of detected and suppressed errors, rerun with: -v
==10173== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

ASan 不能 , valgrind 不能

2. 寫一個簡單程式 with ASan，Stack buffer overflow 剛好越過 redzone(並沒有對 redzone 做讀寫)，並告訴我 ASan 能否找的出來？

```
// 有問題的程式碼
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main()
{
  int a[8] = {0};
  int b[8] = {0};
  a[20] = 10;

  return 0;
}
```

```
ASan report

estee@estee-VirtualBox:~/workspace/SoftwareTest/lab7$ ./S2
```

ASan 不能