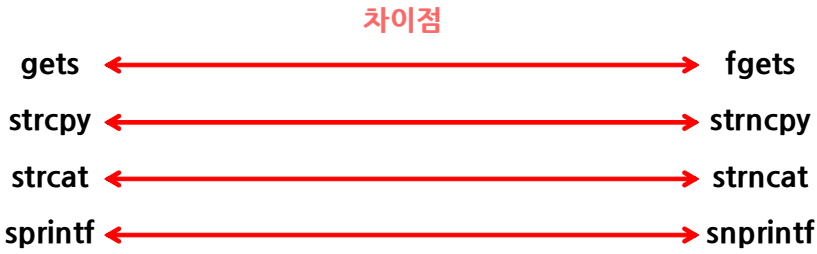
한양대학교 소프트웨어학부
자람 31기 김민우

슬라이드 쇼가 끝났습니다. 끝내려면 마우스를 누릅니다.

취약점



scanf, fscanf, sscanf와 같은 함수는 입력 시 버퍼 최고 길이 명시(ex. “%255s”)

overflow 전

&a	&a+1	&a+2	&a+3	&a+4	&a+5	&a+6	&a+7	&a+8	&a+9
0	1	2	3	4	5	6	7	8	9

&a+10	&a+11	&a+12	&a+13	&a+14	&a+15	&b	&b+1	&b+2	&b+3
10	11	12	13	14	15	A	B	C	D

overflow 후

&a	&a+1	&a+2	&a+3	&a+4	&a+5	&a+6	&a+7	&a+8	&a+9
0	1	2	3	4	5	6	7	8	9

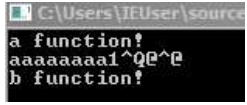
&a+10	&a+11	&a+12	&a+13	&a+14	&a+15	&b	&b+1	&b+2	&b+3
10	11	12	13	14	15	16	17	C	D

32bit

실습 환경

운영체제 : win7
dbg : ollydbg 201
editor : visual studio 2017

```
00401100  r$  55              PUSH EBP                        Project3.a(void)
00401101  .   8BEC            MOV EBP,ESP
00401103  .   51              PUSH ECX
00401104  .   68 34304000     PUSH OFFSET 00403034            [_Format = "a function!@"
00401109  .   E8 42FFFFFF     CALL printf                     printf
0040110E  .   83C4 04         ADD ESP,4
00401111  .   8D45 FC         LEA EAX,[LOCAL.1]
00401114  .   50              PUSH EAX
00401115  .   68 30304000     PUSH OFFSET 00403030            [_Format = "%s"
0040111A  .   E8 A1FFFFFF     CALL scanf                      scanf
0040111F  .   83C4 08         ADD ESP,8
00401122  .   8BE5            MOV ESP,EBP
00401124  .   5D              POP EBP
00401125  L.  C3              RETN
00401126      CC              INT3
00401127      CC              INT3
00401128      CC              INT3
00401129      CC              INT3
0040112A      CC              INT3
0040112B      CC              INT3
0040112C      CC              INT3
0040112D      CC              INT3
0040112E      CC              INT3
0040112F      CC              INT3
00401130  r$  55              PUSH EBP                        Project3.b(void)
00401131  .   8BEC            MOV EBP,ESP
00401133  .   68 20304000     PUSH OFFSET 00403020            [_Format = "b function!@"
00401138  .   E8 13FFFFFF     CALL printf                     printf
0040113D  .   83C4 04         ADD ESP,4
00401140  .   FF15 8C204000   CALL DWORD PTR DS:[<&api-ms-win-crt-run
00401146  .   5D              POP EBP
00401147  L.  C3              RETN
```



주소값은 little endian을 ascii값을 통해 보내줌

64bit

실습 환경

운영체제 : ubuntu 18.04
editor : vim

```
Dump of assembler code for function main:
   0x0000000000000743 <+0>:     push   %rbp
   0x0000000000000744 <+1>:     mov    %rsp,%rbp
   0x0000000000000747 <+4>:     mov    $0x0,%eax
   0x000000000000074c <+9>:     callq  0x6fa <a>
   0x0000000000000751 <+14>:    mov    $0x0,%eax
   0x0000000000000756 <+19>:    pop    %rbp
   0x0000000000000757 <+20>:    retq
End of assembler dump.
(gdb) disas a
Dump of assembler code for function a:
   0x00000000000006fa <+0>:     push   %rbp
   0x00000000000006fb <+1>:     mov    %rsp,%rbp
   0x00000000000006fe <+4>:     sub    $0x10,%rsp
   0x0000000000000702 <+8>:     lea    0xdb(%rip),%rdi        # 0x7e4
   0x0000000000000709 <+15>:    callq  0x5b0 <puts@plt>
   0x000000000000070e <+20>:    lea    -0x4(%rbp),%rax
   0x0000000000000712 <+24>:    mov    %rax,%rsi
   0x0000000000000715 <+27>:    lea    0xd4(%rip),%rdi        # 0x7f0
   0x000000000000071c <+34>:    mov    $0x0,%eax
   0x0000000000000721 <+39>:    callq  0x5c0 <__isoc99_scanf@plt>
   0x0000000000000726 <+44>:    nop
   0x0000000000000727 <+45>:    leaveq
   0x0000000000000728 <+46>:    retq
End of assembler dump.
(gdb) disas b
Dump of assembler code for function b:
   0x0000000000000729 <+0>:     push   %rbp
   0x000000000000072a <+1>:     mov    %rsp,%rbp
   0x000000000000072d <+4>:     lea    0xbf(%rip),%rdi        # 0x7f3
   0x0000000000000734 <+11>:    callq  0x5b0 <puts@plt>
   0x0000000000000739 <+16>:    mov    $0x0,%edi
   0x000000000000073e <+21>:    callq  0x5d0 <exit@plt>
End of assembler dump.
```

대비책

ALSR

메모리 공격을 방지하기 위해 프로세스 생성 시 할당 메모

리 주소를 랜덤으로 할당하는 기능

```
root@kimminwoo-VirtualBox:/home/kimminwoo# cat /proc/self/maps
556280d4e000-556280d56000 r-xp 00000000 08:01 1703961
556280f55000-556280f56000 r--p 00007000 08:01 1703961
556280f56000-556280f57000 rw-p 00008000 08:01 1703961
5562816c9000-5562816ea000 rw-p 00000000 00:00 0
7f3428c90000-7f34297a1000 r--p 00000000 08:01 2366044
7f34297a1000-7f3429988000 r-xp 00000000 08:01 530029
7f3429988000-7f3429b88000 ---p 001e7000 08:01 530029
7f3429b88000-7f3429b8c000 r--p 001e7000 08:01 530029
7f3429b8c000-7f3429b8e000 rw-p 001eb000 08:01 530029
7f3429b8e000-7f3429b92000 rw-p 00000000 00:00 0
7f3429b92000-7f3429bb0000 r-xp 00000000 08:01 530001
7f3429da3000-7f3429da6000 rw-p 00000000 00:00 0
7f3429db9000-7f3429dba000 r--p 00027000 08:01 530001
7f3429dba000-7f3429dbb000 rw-p 00028000 08:01 530001
7f3429dbb000-7f3429dbc000 rw-p 00000000 00:00 0
7ffcf054f000-7ffcf0570000 rw-p 00000000 00:00 0
7ffcf0581000-7ffcf0584000 r--p 00000000 00:00 0
7ffcf0584000-7ffcf0585000 r-xp 00000000 00:00 0
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0
```

```
root@kimminwoo-VirtualBox:/home/kimminwoo# cat /proc/self/maps
560cc34f5000-560cc34fd000 r-xp 00000000 08:01 1703961
560cc36fc000-560cc36fd000 r--p 00007000 08:01 1703961
560cc36fd000-560cc36fe000 rw-p 00008000 08:01 1703961
560cc47d0000-560cc47f1000 rw-p 00000000 00:00 0
7f9265961000-7f9266472000 r--p 00000000 08:01 2366044
7f9266472000-7f9266659000 r-xp 00000000 08:01 530029
7f9266659000-7f9266859000 ---p 001e7000 08:01 530029
7f9266859000-7f926685d000 r--p 001e7000 08:01 530029
7f926685d000-7f926685f000 rw-p 001eb000 08:01 530029
7f926685f000-7f9266863000 rw-p 00000000 00:00 0
7f9266863000-7f9266888a000 r-xp 00000000 08:01 530001
7f9266a53000-7f9266a77000 rw-p 00000000 00:00 0
7f9266a8a000-7f9266a8b000 r--p 00027000 08:01 530001
7f9266a8b000-7f9266a8c000 rw-p 00028000 08:01 530001
7f9266a8c000-7f9266a8d000 rw-p 00000000 00:00 0
7fff611c0000-7fff611e1000 rw-p 00000000 00:00 0
7fff611f0000-7fff611f3000 r--p 00000000 00:00 0
7fff611f3000-7fff611f4000 r-xp 00000000 00:00 0
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0
```

대비책

ASCII-Armor

RET에서 호출하는 라이브러리 함수의 주소값에

0x00을 넣어 삽입된 함수의 호출 방지

```
kimminwoo@kimminwoo-VirtualBox:~/bof$ cat /proc/self/maps
555555554000-55555555c000 r-xp 00000000 08:01 1703961                    /bin/cat
55555575b000-55555575c000 r--p 00007000 08:01 1703961                    /bin/cat
55555575c000-55555575d000 rw-p 00008000 08:01 1703961                    /bin/cat
55555575d000-55555577e000 rw-p 00000000 00:00 0                          [heap]
7ffff6ed3000-7ffff79e4000 r--p 00000000 08:01 2366044                    /usr/lib/locale/locale-archive
7ffff79e4000-7ffff7bcb000 r-xp 000000000 08:01 530029                    /lib/x86_64-linux-gnu/libc-2.27.so
7ffff7bcb000-7ffff7dcb000 ---p 001e7000 08:01 530029                     /lib/x86_64-linux-gnu/libc-2.27.so
7ffff7dcb000-7ffff7dcf000 r--p 001e7000 08:01 530029                     /lib/x86_64-linux-gnu/libc-2.27.so
7ffff7dcf000-7ffff7dd1000 rw-p 001eb000 08:01 530029                     /lib/x86_64-linux-gnu/libc-2.27.so
7ffff7dd1000-7ffff7dd5000 rw-p 00000000 00:00 0
7ffff7dd5000-7ffff7dfc000 r-xp 000000000 08:01 530001                    /lib/x86_64-linux-gnu/ld-2.27.so
7ffff7fc1000-7ffff7fe5000 rw-p 00000000 00:00 0
7ffff7ff8000-7ffff7ffb000 r--p 00000000 00:00 0                          [vvar]
7ffff7ffb000-7ffff7ffc000 r-xp 00000000 00:00 0                          [vdso]
7ffff7ffc000-7ffff7ffd000 r--p 000027000 08:01 530001                    /lib/x86_64-linux-gnu/ld-2.27.so
7ffff7ffd000-7ffff7ffe000 rw-p 000028000 08:01 530001                    /lib/x86_64-linux-gnu/ld-2.27.so
7ffff7ffe000-7ffff7fff000 rw-p 00000000 00:00 0
7ffffffde000-7ffffffff000 rw-p 00000000 00:00 0                          [stack]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
```

대비책

Canary

변수의 주소 지정 시 변수 메모리 부분과 SFP, RET사이에
canary라는 임의의값을 통해 buffer overflow 감지

RET은 함수의 실행 이후 return되는 address

대비책

Canary

변수의 주소 지정 시 변수 메모리 부분과 SFP, RET사이에

```
kimminwoo@kimminwoo-VirtualBox:~/bof$ (python -c 'print "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x59\x53\x89\xe1\x89\xc2\xb0\x0b\xcd\x80\x31\xc0\xb0\x01\xcd\x80"+"\x90"*237+"\xc0\xfd\xff\xbf"';cat)|./dep
Input: 0x7fffffffddc0
*** stack smashing detected ***: <unknown> terminated

Aborted (core dumped)
```

RET은 함수의 실행 이후 return되는 address

대비책

DEP(NX)

데이터 영역에서 특정 코드의 실행을 방지하는 기능

gcc -z execstack {filename}.c -o {filename}으로 실행권을 부여 가능

checksec(.sh)로 확인가능

```
kimminwoo@kimminwoo-VirtualBox:~/bof$ checksec -f stack
RELRO           STACK CANARY      NX                PIE
Full RELRO      No canary found   [redacted]        PIE enabled
```

대비책

RELRO

ELF 바이너리 또는 프로세스의 데이터 섹션을 보호하는 기술

이해하기 위한 개념 : Lazy Binding, GOT Overwirte

PIE

전체가 위치 독립 코드로 이루어진 실행 가능한 바이너리

출처 : https://bpsecblog.wordpress.com/memory_protect_linux/