packer

Medium | Reverse Engineering | picoCTF 2024 | browser_webshell_solvable

AUTHOR: MUBARAK MIKAIL

Description

Reverse this linux executable?

binary

Hints

1

What can we do to reduce the size of a binary after compiling it.

```
packer % file out
out: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, no section
 header
packer %
```

Ketika di strings / cat muncul hint bahwa file out ini di packing dengan upx

```
PROT_EXEC|PROT_WRITE failed.
$Info: This file is packed with the UPX executable packer http://upx.sf.net $
$Id: UPX 3.95 Copyright (C) 1996-2018 the UPX Team. All Rights Reserved. $
```

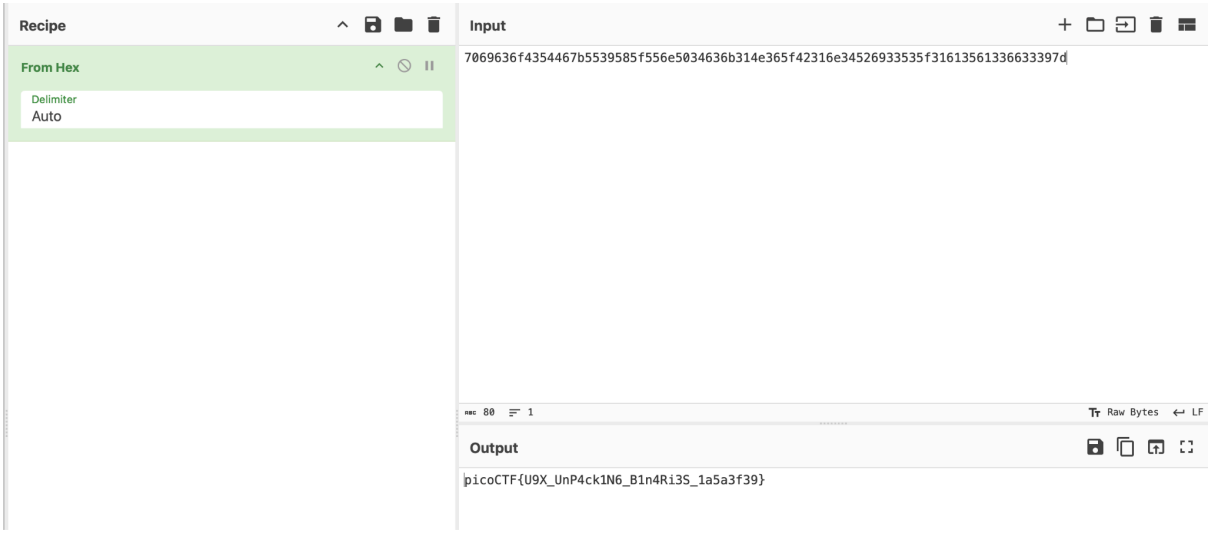
Langsung saja kita download tools packernya di mac dengan command brew install upx.

```
packer % ls
out
packer % upx -d out -o out_decompressed
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2026
UPX 5.1.0       Markus Oberhumer, Laszlo Molnar & John Reiser    Jan 7th 2026

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
[WARNING] bad b_info at 0x4b718

[WARNING] ... recovery at 0x4b714

    872088 <-    336520   38.59%   linux/amd64   out_decompressed

Unpacked 1 file.
packer % ls
out                     out_decompressed
packer % file out
out: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, no section
 header
packer % file out_decompressed
out_decompressed: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linke
d, BuildID[sha1]=1c5ee6208dac5576d6893e662951fa6f35e49efc, for GNU/Linux 3.2.0, not stripped
packer %
```

Setelah diinstal dan di decompress dengan command seperti di atas, aku mendapatkan file yang dapat dianalisis dengan binary ninja.

```
int64_t main()

    void var_a8
    void* i = &var_a8
    void* fsbase
    int64_t rax = *(fsbase + 0x28)
    int64_t var_a0 = 0x64
    int64_t var_98 = 0x63
    int64_t rax_3 = divu.dp.q(0:0x73, 0x10) * 0x10

    while (i != &var_a8 - (rax_3 & 0xfffffffffffff000))
        i -= 0x1000
        *(i + 0xff8) = *(i + 0xff8)

    char* rsp = i - (zx.q(rax_3.d) & 0xfff)

    if ((zx.q(rax_3.d) & 0xfff) != 0)
        int64_t* rax_6 = (zx.q(rax_3.d) & 0xfff) - 8 + rsp
        *rax_6 = *rax_6

    int64_t var_88
    __builtin_strncpy(dest: &var_88,
        src: "7069636f4354467b5539585f556e5034636b314e365f42316e34526933535f31613561336633397d",
        count: 0x64)
    _IO_printf("Enter the password to unlock this file: ", 0)
    _IO_fgets(rsp, var_a0.d, stdin)
    _IO_printf("You entered: %s\n", 0)

    if (sub_4010d0(rsp, &var_88, var_a0, &var_88) != 0)
        _IO_puts("Access denied")
    else
        _IO_puts("Password correct, please see flag: "
        "7069636f4354467b5539585f556e5034636b314e365f42316e34526933535f31613561336633397d")
        _IO_puts(&var_88)

    if (rax == *(fsbase + 0x28))
        return 0

    __stack_chk_fail()
    noreturn
```

Ternyata ada flag yang di encode, kita coba decode dengan cyberchef.



Flag : picoCTF{U9X_UnP4ck1N6_B1n4Ri3S_1a5a3f39}