TIM COD MUSANG

Write-up: Web CTF – Akane

1. Analisis Challenge

- **Nama soal**: Akane
- **Kategori**: Web Exploitation
- **Author**: aimardcr

Deskripsi challenge menyebutkan bahwa *flag* disimpan di **environment variable**. Peserta juga diberikan attachment Akane_dist.zip yang berisi source code service, serta setiap instance hanya hidup 15 menit.

Dari analisis file dan percobaan akses, terlihat adanya mekanisme **debug header** yang bisa digunakan untuk melakukan *out-of-bound (OOB) argv leak*, yaitu meloloskan data sensitif (termasuk environment variable) dengan mengontrol nilai X-Debug-Index.

---

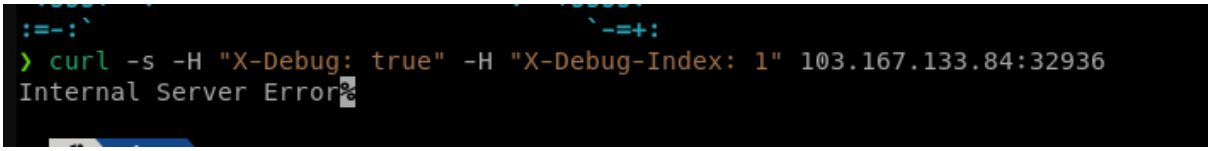
2. Trik Penting

- Service menerima header khusus:
    - X-Debug: true
    - X-Debug-Index: <angka>
- Dengan mengatur index tertentu, kita bisa membaca bagian dari internal array (argv/env).
- Index yang salah → Internal Server Error.
- Index yang tepat → *leak* environment variable.
- Trik ini memanfaatkan **Out-of-Bound Read** untuk mengakses data sensitif.

---

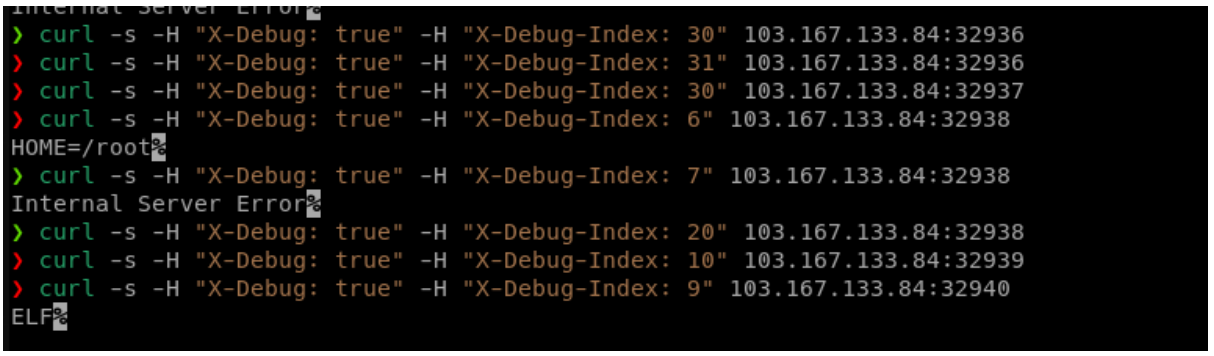
3. Eksploitasi

1. Uji beberapa index dengan curl:



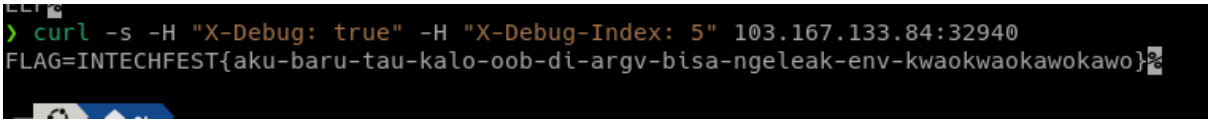
→ menghasilkan error.

2. Coba index lain (misalnya 30, 31, 58, 4, dst) di port **32891**:



→Masih mengalami error, dan semisal mencoba mengecek index diatas 9 maka server akan mati

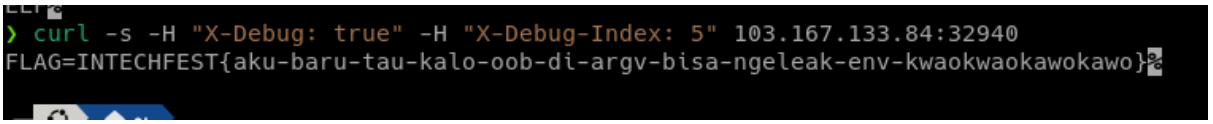
3. Akhirnya ditemukan index yang mengeluarkan **environment variable berisi FLAG**:



---

4. Hasil

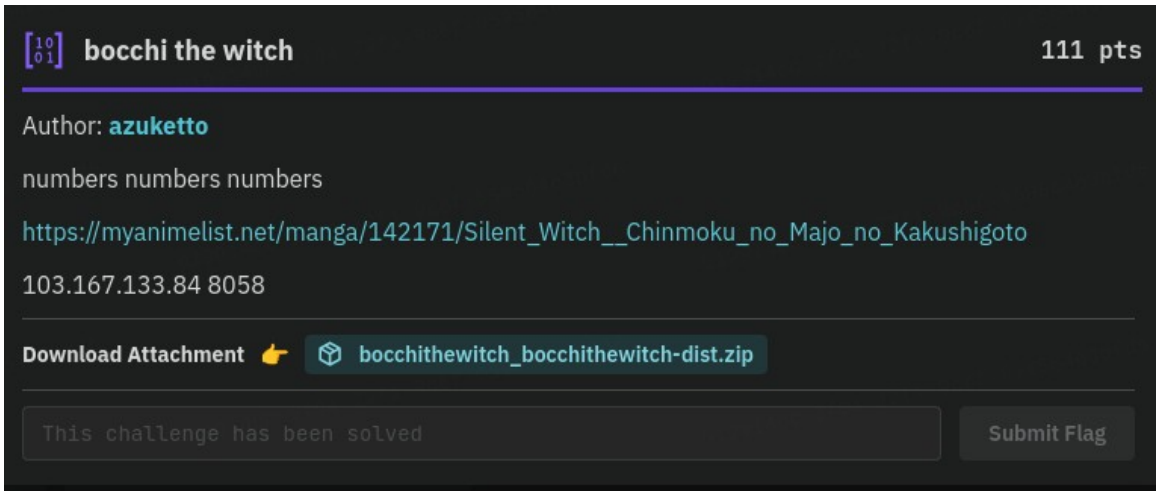
Flag berhasil ter-*leak* dari environment variable:



5. FLAG

INTECHFEST{aku-baru-tau-kalo-oob-di-argv-bisa-ngeleak-env-kwaokwaokwaokawokawo}

Write-up: Crypto CTF – bocchi the witch



1. Analisis Challenge

- Server memberikan modulus RSA (N), public exponent e = 65537, beberapa ciphertext (ct_i), dan sebuah “Hint”.
- “Hint” sebenarnya adalah hasil dari operasi (e^{-1} mod d) yang di-*mask* dengan operasi bitwise AND.
- Kita bisa mengirim nilai mask, dan server akan mengembalikan hasil (e^{-1} mod d) & mask.
- Normalnya hasilnya terbatas karena aturan bit_count, tapi ada trik yang bisa digunakan.

2. Trik Penting

Di Python, `(-1).bit_count() == 1`. Jika kita mengirim `mask = -1`, maka:
- Operasi x & -1 menghasilkan x, sehingga server akan memberikan nilai penuh dari de = e^{-1} mod d.
- Biasanya hanya sebagian nilai yang diberikan, tapi dengan trik ini seluruh de bisa didapat.

3. Cara Dapatkan d

Dengan nilai de di tangan, kita bisa gunakan rumus:

$e * de = 1 + k \cdot d$

Karena e kecil (65537), kita brute-force k dari 1 sampai e. Jika (e*de - 1) habis dibagi k, maka d = (e*de - 1)/k. Kandidat d ini bisa diuji untuk membangun private key RSA.

4. Decrypt Ciphertext

Setelah menemukan (n, e, d), kita dapat membangun private key RSA. Kemudian gunakan private key untuk mendekripsi salah satu ciphertext (ct_0). Plaintext hasil dekripsi adalah angka s.
Kirim s kembali ke server sebagai guess, dan server akan memberikan flag.

5. Solver Script

Berikut contoh solver (Python) yang otomatis melakukan semua langkah di atas:

```
import socket, re
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP

HOST, PORT = "103.167.133.84", 8058
E = 65537

with socket.create_connection((HOST, PORT)) as sock:
    sock.recv(4096)
    sock.sendall(b"-1\n")
    data = sock.recv(16384).decode()

    n = int(re.search(r"N: (\d+)", data).group(1))
    de = int(re.search(r"Hint: (\d+)", data).group(1))
    ct_hex = re.findall(r"ct_\d+: ([0-9a-f]+)", data)[0]
    ct = bytes.fromhex(ct_hex)

    t = E * de - 1
    d = None
    for k in range(1, E):
        if t % k == 0:
            cand = t // k
            try:
                key = RSA.construct((n, E, cand))
                pt = PKCS1_OAEP.new(key).decrypt(ct)
                s = int.from_bytes(pt, "big")
                d = cand
                break
            except: pass

    sock.sendall(f"{s}\n".encode())
    print(sock.recv(4096).decode())
```

FLAG:

INTECHFEST{why_am_i_still_creating_challs_I_wanna_read_novels_be09a5d6e2ed328a}