

Write Up CTF Find IT! 2025

Made with love by Cimahi x Kacapiring:

Wisa Ahmaduta Dinutama

Muhammad Aidan Fathullah

Naufarrel Zhafif Abhista



Intro tipis:

Our first capture the flag competition! Jujur kurang persiapan karena keos di tengah serangan tugas besar yang gak ada habisnya. But here we are! Berhasil survive 25 jam dengan nge-solve total 8 soal and manage to secure the 25th rank position (sebelum leaderboard di-freeze). Huge thanks buat pihak UGM yang udah mau ngadain acara lomba ctf ini yang relatif beginner friendly buat kami para pemain baru.

I. Miscellaneous

A. Absen

Challenge

111 Solves

×

Absen

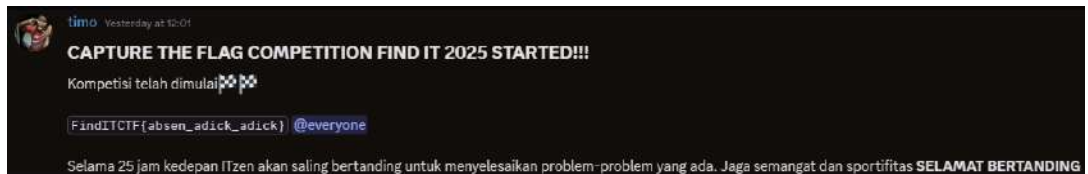
100

ayok absen sebelum marathon ctf

Flag

Submit

Kita absen dulu ya adik-adik. Untuk solve challenge ini, kita cukup memasukkan flag yang sudah disediakan oleh panitia via discord ke website ctf.find-it.id.



FindITCTF{absen_adick_adick}

B. cek-cek

Challenge

60 Solves

×

cek-cek

100

Hei, aku baru belajar python. Semoga aku tidak melupakan sesuatu.

author: [hilmo](#)

[nc ctf.find-it.id 7001](#)

 [main.py](#)

1. Analisis kode [main.py](#) yang diberikan terlebih dahulu

```
import hashlib
import os

from secret import FLAG

def check(s):
    if "." in s or "flag" in s: #mengecek "." dan "flag" pada s
        return False
    return True

hash_obj = hashlib.blake2b()
hash_obj.update(FLAG.encode())
flag = hash_obj.hexdigest() # menyimpan flag dalam bentuk hashed

def open_file(file_name):
    if not check(file_name): #membuka file dan melakukan checking
```

```

        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: ") # meminta nama file jika 1
            print(open_file(file_name)) # mencetak file yang diminta
        elif choice == "2":
            print("ok, here the flag:")
            print(flag) # mengembalikan hashed flag jika 2
        else:
            print("invalid choice")

```

2. Setelah melakukan analisis, kita ketahui bahwa jika input 2 maka sistem akan mengembalikan hashed flag. Tetapi, karena hash merupakan prosedur matematika yang irreversible sehingga hal ini sebenarnya tidak terlalu berguna.

3. Oleh karena itu, kita menggunakan pendekatan kedua, yaitu mencoba mendapatkan file flag tanpa menggunakan substring "." dan "flag" (memilih opsi 1).
4. Hubungkan perangkat dengan server challenge dengan menggunakan command "nc ctf.find-it.id 7001" pada terminal.
5. Ketika sudah masuk, akan muncul prompt seperti yang tertera pada kode [main.py](#). Pilih opsi 1.

```
(wisa@LAPTOP-A3FF01BR)-[~]  
$ nc ctf.find-it.id 7001  
Do you want check my file?  
1. yes  
2. no  
>>> 1
```

6. Kita cari cara untuk mencetak file. Hal tersebut ternyata dapat dilakukan dengan memanfaatkan file descriptor yang dimiliki linux. Pada direktori /proc/self terdapat symbolic links berupa angka-angka yang merujuk ke file tertentu. Proses mendapatkan flag dapat digunakan dengan menebak angka mulai dari 0.
7. Lakukan iterasi proses 6 berulang kali sampai mendapatkan flag akhir.

```
file name: /proc/self/fd/4  
error bang  
Do you want check my file?  
1. yes  
2. no  
>>> 1  
file name: /proc/self/fd/5  
FindITCTF{cl0s3_y0ur_f1l3s_1mmed14t3ly_0r_w0w0_w1ll_f1nd_y0u}
```

FindITCTF{cl0s3_y0ur_f1l3s_1mmed14t3ly_0r_w0w0_w1ll_f1nd_y0u}

C. 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...

Submit

1. Pada challenge ini, kita diberikan gambar “mleyot”



2. Hal yang harus kita lakukan adalah melakukan rekonstruksi gambar sebelum “mleyot” agar kita bisa mengetahui lokasinya.
3. Untungnya, deskripsi yang diberikan sudah lebih dari cukup untuk merekonstruksi gambar asli sebelum terdistorsi. Hal pertama yang perlu dilakukan adalah membuat python script dengan library pillow untuk merekonstruksi gambar dengan melakukan pemindahan piksel. Adapun script yang dibuat adalah sebagai berikut.

```
from PIL import Image

# Buka gambar
img = Image.open("distorted.png")
width, height = img.size

# Buat gambar baru untuk hasil perbaikan
fixed_img = Image.new("RGB", (width, height))

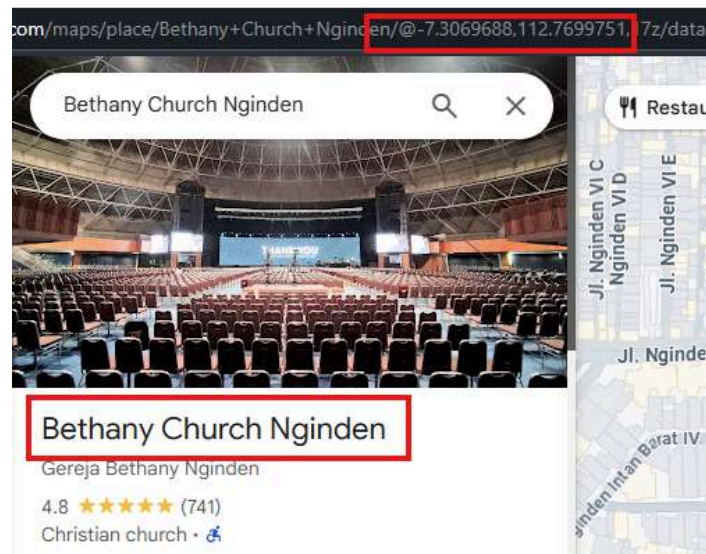
for y in range(height):
    # Hitung jumlah shift (misalnya 5 piksel per baris)
    shift = (y * 5) % width
    # Ambil satu baris
    row = img.crop((0, y, width, y + 1))
    # Geser ke kiri (kebalikan dari distorsi)
    fixed_row = Image.new("RGB", (width, 1))
    fixed_row.paste(row.crop((shift, 0, width, 1)), (0, 0))
    fixed_row.paste(row.crop((0, 0, shift, 1)), (width - shift, 0))
    # Tempel ke gambar baru
    fixed_img.paste(fixed_row, (0, y))

# Simpan hasilnya
fixed_img.save("fixed.png")
```

4. Jalankan kode python tersebut terhadap gambar sebelumnya dan kita akan mendapatkan gambar asli.



5. Gunakan google image search untuk mencari lokasi gambar. Setelah pencarian, kita akan mengetahui bahwa lokasi pada gambar adalah Gereja Bethany Nginden. Setelah itu, catat nama, lintang, dan bujur untuk flag yang dibutuhkan.



FindITCTF{-7.3068_112.7725_Gereja_Bethany_Nginden}

II. Reverse

A. XOR Madness

Challenge

107 Solves


×

xor_madness

100

Bombombini Gusini adalah seorang mahasiswa tahun pertama jurusan Teknologi Informasi yang tengah mendalami cryptography dan malware analysis di mata kuliah Peretasan Beretika. Suatu hari, dosen memberikan tugas berupa sebuah binary file bernama xor_madness.bin. Katanya jika ia berhasil mendapatkan "sesuatu" dari binary file tersebut, maka ia akan langsung mendapatkan nilai A. Bantulah ia untuk bisa mendapatkan "sesuatu" tersebut.

author: [mojitodev](#)

 xor_madn...

Flag

Submit

1. Pada challenge ini, kita diberikan sebuah file binary yang harus kita decrypt untuk mendapatkan flag. Adapun isi dari file tersebut adalah sebagai berikut.

```
Uz}wZGPGUhzzj'Lq } aL"}"Lu 'tL}j'Lq'}tn
```

2. Hal ini dapat dilakukan dengan mencoba algoritma dekripsi yang berhubungan dengan xor. Untuk melakukan hal tersebut, terdapat banyak alternatif seperti membuat script atau menggunakan tools. Namun, pada

solusi kali ini, kami memutuskan untuk menggunakan tools pada sebuah website bernama <https://www.dcode.fr/xor-cipher>.

3. Masukkan kode yang terdapat di dalam file binary ke dalam website dan coba bereksperimen dengan pilihan yang ada. Ternyata, flag ditemukan ketika set ukuran key menjadi 1 byte.

The screenshot shows the DCode.fr website interface for the XOR Cipher tool. On the left, a sidebar titled 'Search for a tool' displays search results for the keyword 'FindITCTF{iy4_b3n3r_1n1_fl4g_ny4_b4ng}', which is highlighted with a red box. The main content area is titled 'XOR CIPHER' and 'XOR DECODER'. It features a text input field containing the same string, a dropdown menu for 'ENCRIPTION/DECRYPTION METHOD' set to 'KNOWING THE KEY SIZE (IN BYTES)', and a 'RESULTS FORMAT' dropdown set to 'String of Unicode Characters (UTF-8)'. The 'KEY SIZE (IN BYTES)' is set to 1. The 'ENCRYPT / DECRYPT' button is visible. Below the main tool, there is a 'CRYPTANALYSIS' section with a 'SEARCH FOR KEY SIZE (IN BYTES)' button and an 'ANALYZE' button. At the bottom, there is an 'XOR CALCULATOR' section with a 'BINARY NUMBER/MESSAGE 1' input field.

FindITCTF{iy4_b3n3r_1n1_fl4g_ny4_b4ng}

III. Cryptography

A. caesar cipher

Challenge

114 Solves

×

caesar cipher

100

author: mojitodev

Pada suatu malam, Tung Tung Tung Tung Sahur ingin mendatangi seorang pemuda yang tidak bangun sahur setelah dipanggil sahur sebanyak 3 kali, tetapi tidak nyaut. Masalahnya adalah pintu kamar pemuda tersebut terkunci dengan password tertentu, tetapi terdapat file `cipher.txt` yang tersimpan dalam flashdisk di dekatnya yang bisa digunakan untuk menemukan passwordnya. Bantulah Tung Tung Tung Tung sahur untuk menemukan passwordnya!

author: mojitodev

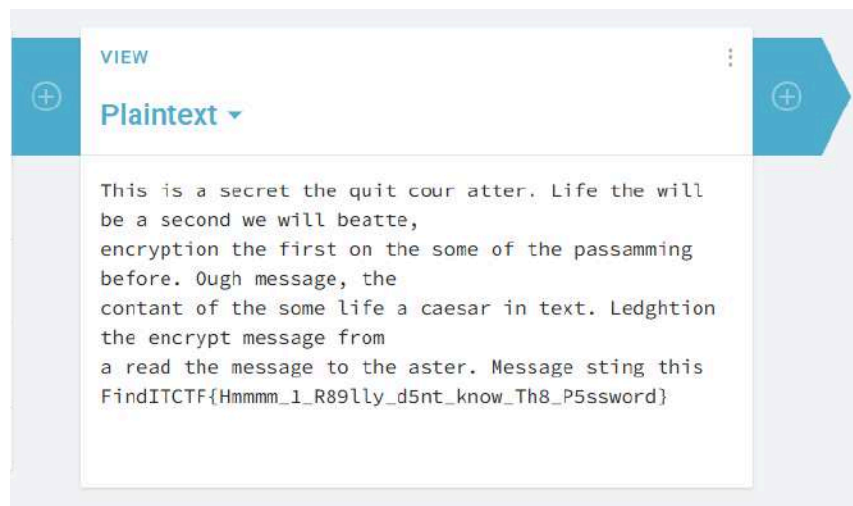
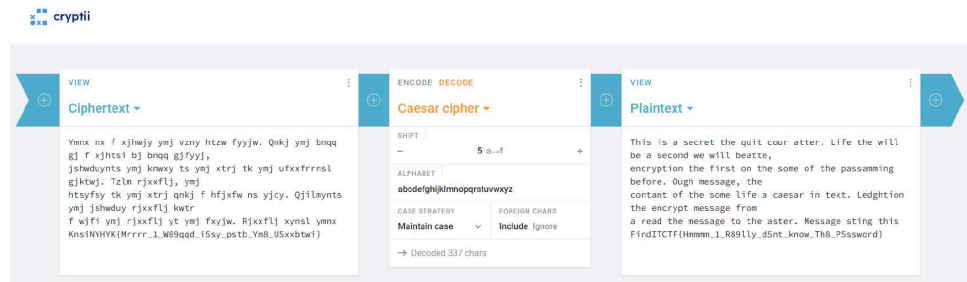
 ciphertext...

Flag

Submit

1. Caesar cipher merupakan algoritma enkripsi sederhana yang menukar setiap huruf dengan posisi lain di alfabet.
2. Proses decoding dapat dilakukan dengan mudah menggunakan tools dekriptor online. Dalam solusi kali ini, kami memutuskan untuk menggunakan website <https://cryptii.com/pipes/caesar-cipher>.

3. Masukkan ciphertext ke dalam decoder. Kemudian, eksperimen dengan jumlah shift. Setelah beberapa iterasi, flag ditemukan ketika shift bernilai 5.



FindITCTF{Hmmm_1_R89lly_d5nt_know_Th8_P5ssword}

IV. Forensic

A. Oversharing

Challenge

25 Solves

×

Oversharing


775

author: BerlianGabriel

Yo man wassup,

I am so excited, finally passed my probation. Just got assigned to this new high impact project. The IT guy just gave me my account for the project, check out the chat! Can't wait to login and show my worth :D

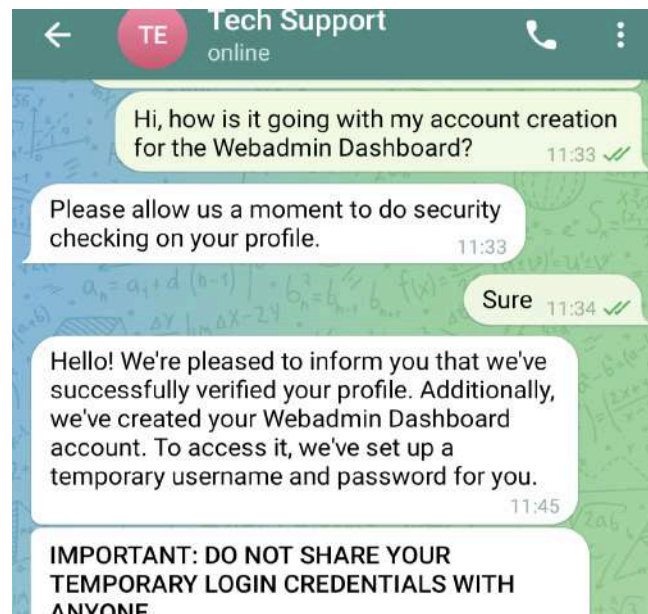
author: [BerlianGabriel](#)

 Overshari...

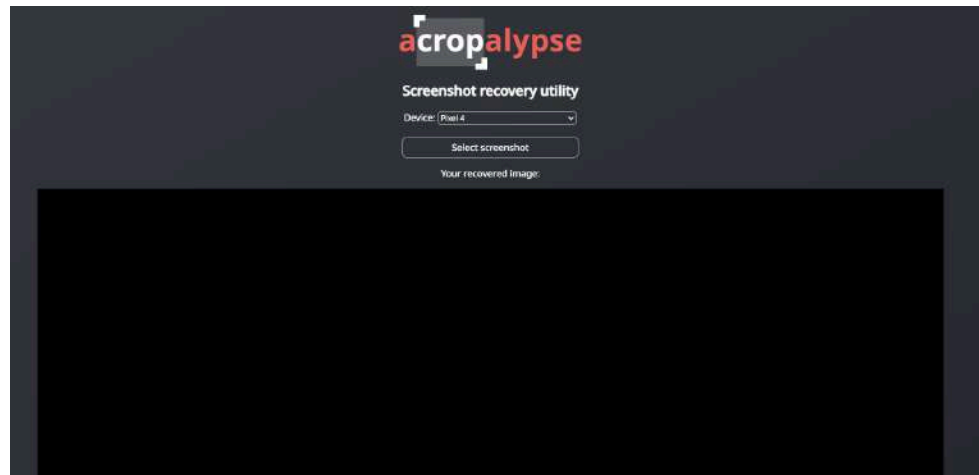
Flag

Submit

1. Pada challenge ini, kita diberikan sebuah gambar sebagai berikut.



2. Pada gambar, dapat dilihat bahwa gambar di-crop sedemikian rupa sehingga login credentials yang ingin dilihat terpotong pada bagian bawah.
3. Hal pertama yang kami curigai adalah bahwa gambar tersebut merupakan gambar dengan [aCropalypse](#) bug, sebuah vulnerability pada Markup, aplikasi screenshot editor pada Google Pixel. Vulnerability tersebut mengakibatkan gambar hasil crop tetap menjaga informasi bagian yang seharusnya tercrop.
4. Untuk menyelesaikan challenge ini, kita bisa menggunakan [acropalypse](#), sebuah website screenshot recovery yang khusus dibuat untuk aCropalypse bug. Terdapat beberapa perangkat yang tersedia pada website. Eksperimen dengan berbagai jenis perangkat pilihan sampai mendapatkan flag.
5. Setelah beberapa iterasi, flag ditemukan pada pilihan perangkat Pixel 4.



FindITCTF{CVE-2023-21036_Hk3MQu1gR3}

V. OSINT

A. 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}

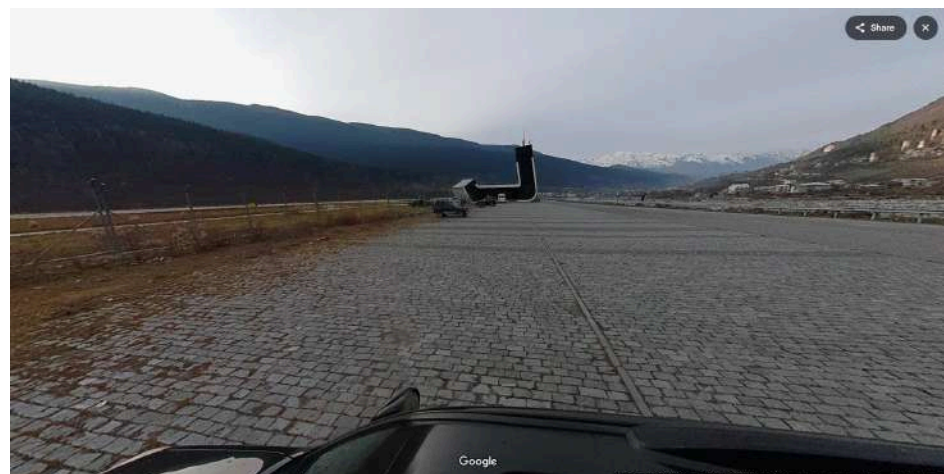
author: **hilmo**

 street_vie...

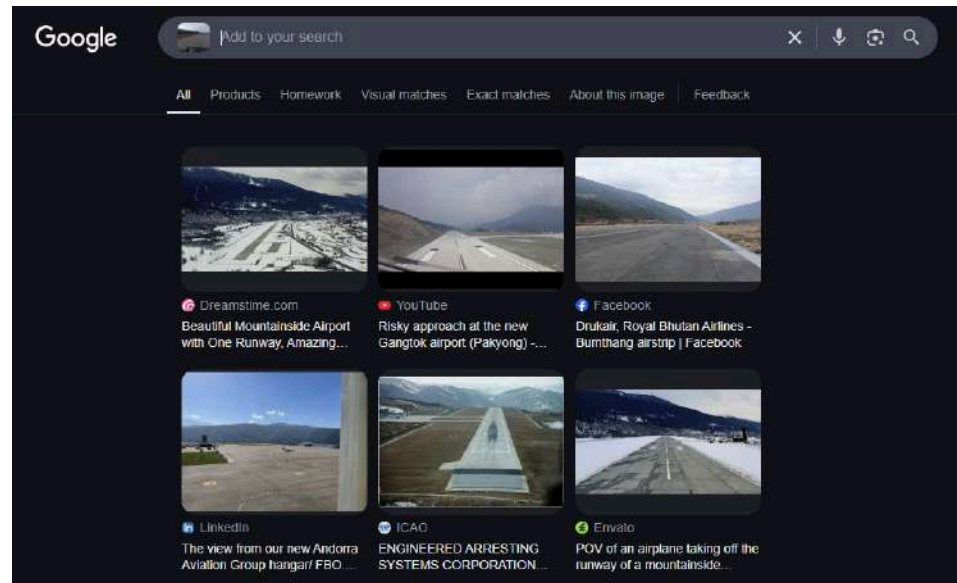
Flag

Submit

1. Pada challenge ini kita diberikan sebuah gambar yang diambil dari google street view.



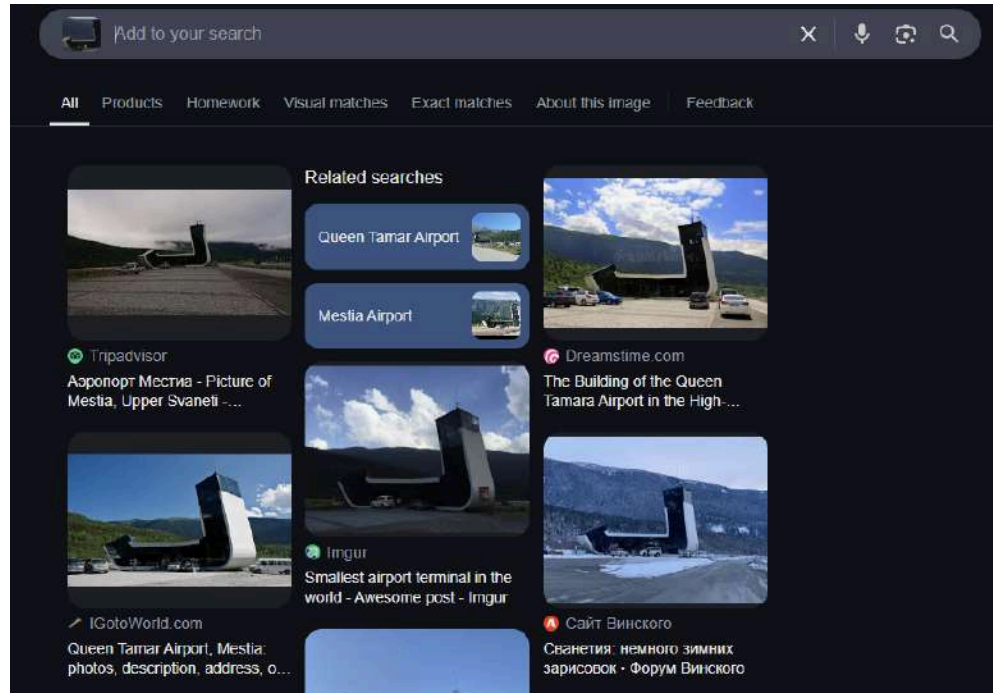
2. Hal pertama yang terlintas pada pikiran kami adalah untuk mencarinya via google image search. Namun, sayangnya pencarian tersebut tidak menghasilkan hasil yang menarik.



3. Setelah itu, kami mencoba pendekatan lain. Kami mencoba menge-crop gambar untuk mendapatkan foto ikonik dari gambar ini, yaitu struktur aneh pada ujung gambar.

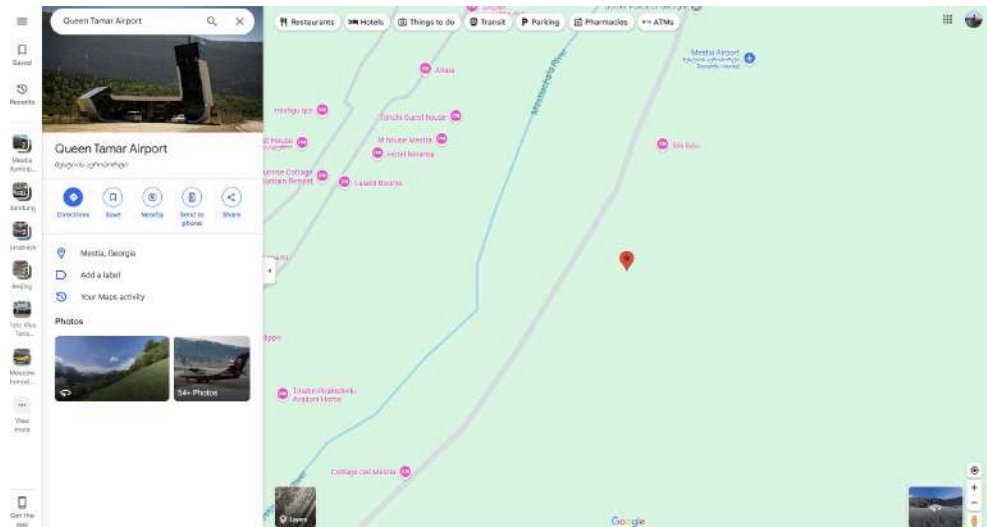


4. Setelah menggunakan gambar yang sudah ter-crop tersebut, barulah kami mendapatkan hasil yang menarik.



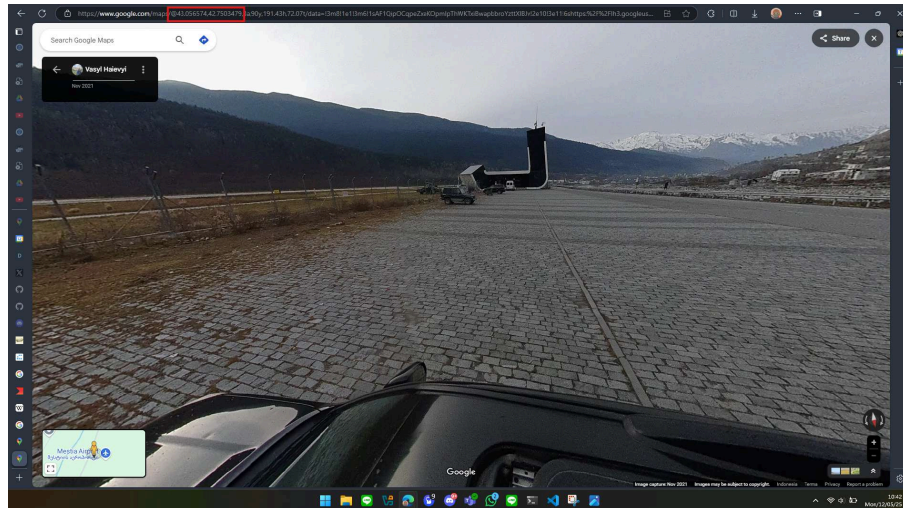
Ternyata foto tersebut diambil di Queen Tamar Airport, bandara di Georgia dengan terminal terkecil di dunia.

5. Kemudian, kami membuka laman Queen Tamar Airport di google maps.

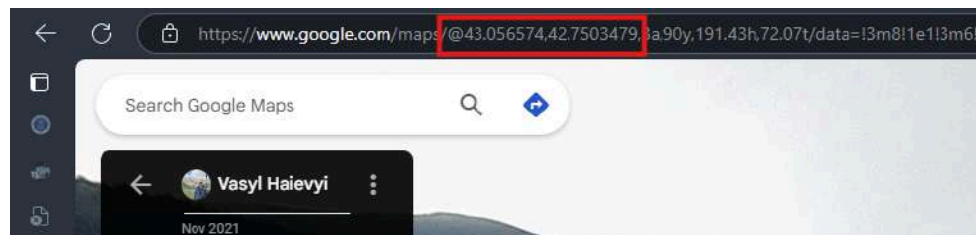


6. Kami sadar, berbeda dengan challenge sebelumnya, koordinat yang diberikan pada flag harus koordinat eksak dari lokasi gambar google street view yang diberikan. Oleh karena itu, kami masuk ke google

street view dan mencoba mencocokkan posisi sedemikian rupa agar persis dengan gambar yang diberikan. Setelah beberapa menit maju mundur, akhirnya kami menemukan lokasi eksak foto google street view tersebut diambil.



7. Kemudian, kita bisa ekstrak koordinatnya dari url google street view tersebut. Perlu diingat bahwa urutan koordinat geografis adalah latitude (lintang), longitude (bujur). Sedangkan, latitude menyimbolkan jarak ke utara atau selatan yang berarti koordinat Y, sehingga longitude merupakan koordinat X. Oleh karena itu, pada flag, masukan angka kedua pada url terlebih dahulu. Setelah itu, baru masukkan angka pertama.



FindITCTF{42.7503479_43.056574}

VI. Web

A. Simple Heist

Challenge

53 Solves

×

Simple Heist

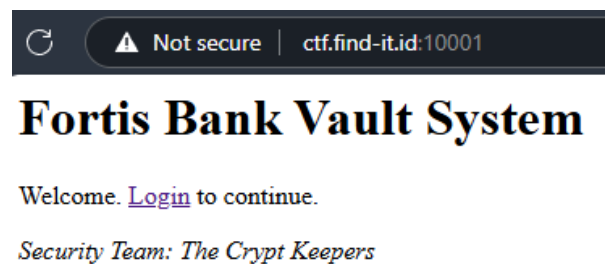
100

gampang sekali, tinggal cari kunci dari brankasnya
cuma internal yang boleh tau banyak hal
author: [hilmios](#)

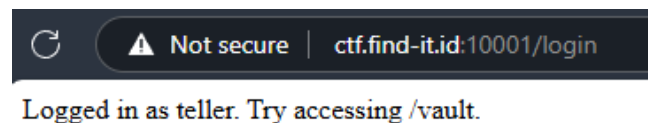
<http://ctf.find-it.id:10001>

Submit

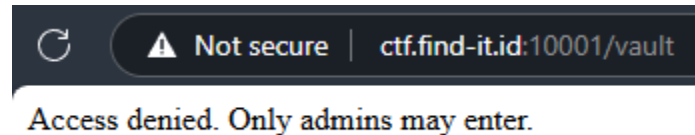
1. Setelah url tersebut dibuka, muncul laman sebagai berikut.



2. Setelah klik login, kita secara otomatis login sebagai teller.



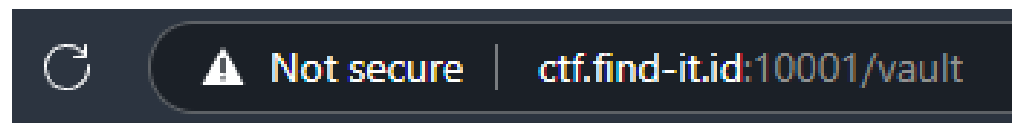
3. Sayangnya, setelah mengganti url dengan path /vault, akses ditolak sebagaimana tertera pada cuplikan laman di bawah ini.



4. Setelah melakukan inspeksi dengan developer tools, tidak ada komponen lain selain HTML. Kami langsung curiga bahwa challenge ini memiliki kaitan dengan cookies. Benar saja, setelah melihat bagian cookies, kami memiliki cookies seperti ini.

| Name | Value |
|---------|--|
| auth | "user:teller bank:Fortis Bank" |
| session | 05f05e21-67c9-4331-abe7-fe905f81cf66.zMWq2kjrz1V4sRfaV8N7_733_Y |
| sig | 7a91f28871e4b9a78f12ff523f068806d6270aaa418fb2a842135faa68843266 |

5. Ide pertama adalah untuk mengganti nilai user menjadi admin, dengan harapan dapat membuka vault. Namun, setelah melakukan hal tersebut justru server mendeteksi bahwa terdapat *tampering* pada cookies.

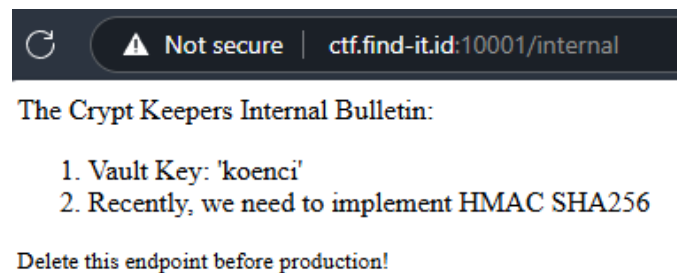


6. Setelah itu kami sadar bahwa server melakukan otentikasi dengan mengecek apakah cookies auth memiliki nilai hash yang sama dengan sig (signature). Awalnya, kami bingung mau bagaimana. Kami mencoba menyalin signature ke suatu website untuk mencari kemungkinan algoritma hash kriptografis apa yang digunakan. Namun, sekalipun kami mendapatkan algoritma

yang diberikan kami tetap tidak bisa menentukan kunci yang digunakan.

7. Ternyata, kami melewati satu clue penting dari deskripsi soal, "cuma internal yang boleh tau banyak hal"

Salah satu dari kami langsung mencoba mengakses ctf.find-it.id:10001/internal untuk menguji hipotesisnya. Benar saja, ternyata laman tersebut menghasilkan page sebagai berikut.



8. Selanjutnya, kami mencoba mengonfirmasi implementasi hashing kami dengan membuktikan apakah hasil hash cookies auth dengan nilai user teller sama dengan sig yang diberikan via website.

Generate HMAC Authentication Code

Enter Plain Text to Compute Hash

usertellerbankFortis Bank

Enter the Secret Key

koenci

Select Cryptographic Hash Function

SHA-256

Output Text Format ☒ Hex ☐ Base64

Compute

Hashed Output:

7a91f28871e4b9a78f12ff523f068806d6270aaa418fb2a842135faa68843266

Ternyata sama!

9. Setelah itu kami masukkan user bernilai admin pada hash generator tersebut dan mendapatkan nilai signature sebagai berikut.

Generate HMAC Authentication Code

Enter Plain Text to Compute Hash

user:admin|bank:Fortis Bank

Enter the Secret Key

koenci

Select Cryptographic Hash Function ?

SHA-256

Output Text Format ☒ Hex ☐ Base64

Compute

Hashed Output:

7f5976dc018b18b360aad2d4c5b3efe099db2bbba363bad5c1932b137f41ba

10. Selanjutnya, kami kembali ke laman vault tadi. Kali ini, kami mengganti auth dengan user:admin dan sig dengan nilai hasil hashing yang sudah didapat.

| Name | Value |
|---------|--|
| auth | "user:admin bank:Fortis Bank" |
| session | 05f05e21-67c9-4331-abe7-fe905f81cf66.zMWq2kjr1V4sRfaV8N7_733_Y |
| sig | 7f5976dc018b18b360aad2d4c5b3efe099db2bbba363bad5c1932b137f41ba |

11. Refresh page dan akhirnya flag berhasil didapatkan!

⌂⚠ Not secure | ctf.find-it.id:10001/vault

Welcome to the vault, admin!
Flag: FindITCTF{BEtEc_10_&1J!)

FindITCTF{BEtEc_10_&1J!)

B. Pixel Plaza

Challenge

32 Solves

×

PixelPlaza


655

author: BerlianGabriel

I'm a consultant, but my client is using a new technology I'm not familiar with. Can I outsource this whitebox pentest project to you?

author: [BerlianGabriel](#)

<http://ctf.find-it.id:6001>

 [main.go](#)

Submit

1. Analisis [main.go](#) yang diberikan terlebih dahulu.

```
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))]) // mencetak quotes
random dari list of quotes yang tersedia
}

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

func apiGuestbook(w http.ResponseWriter, r *http.Request) { // API untuk
menerima input dari user dan menambahkannya ke guest book
    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) { // handler untuk
mengembalikan foto banner
    http.ServeFile(w, nil, "../docs/banner.png")
}

func staticHandler(w http.ResponseWriter, r *http.Request) { // static
handler untuk mengambil resource dari file system
    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)
}

func main() {
    rand.Seed(time.Now().UnixNano())
    mux := http.NewServeMux()
    mux.HandleFunc("/banner.png", banner) // endpoint banner.png memanggil
fungsi banner
    mux.HandleFunc("/api/quote", apiQuote) // endpoint /api/quote
mengembalikan random quote

```

```

mux.HandleFunc("/api/clock", apiClock) // endpoint clock mengembalikan
waktu saat ini
mux.HandleFunc("/api/guestbook", apiGuestbook) // endpoint guestbook
mengembalikan guestbook saat ini
fileServer := http.FileServer(http.FS(webFS))
mux.Handle("/static/", http.StripPrefix("/static/", fileServer)) // api
static untuk mencari file di direktori static
mux.HandleFunc("/", staticHandler) // endpoint / memanggil staticHandler
http.ListenAndServe(":80", mux)
}

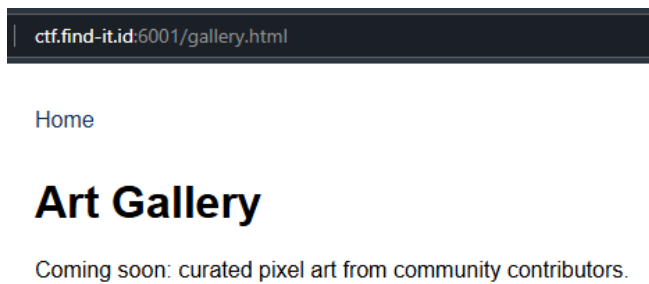
```

- Setelah selesai menganalisis endpoint, kami coba membuka websitenya. Berikut adalah page utama (/). Sebuah banner, navbar, random quote, waktu saat ini, dan guest book.

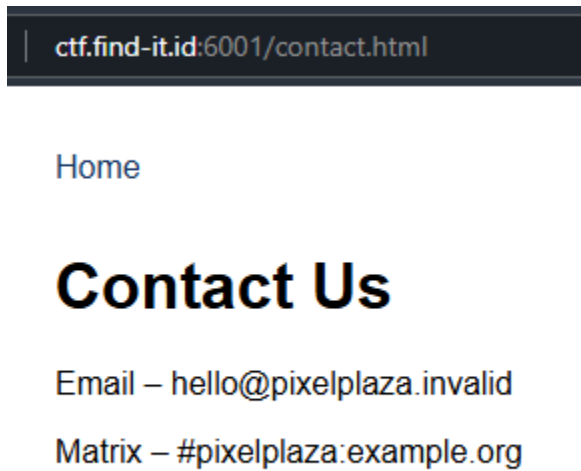


- Berikut juga laman-laman lain setelah mengeklik anchor tag yang tersedia pada laman utama.

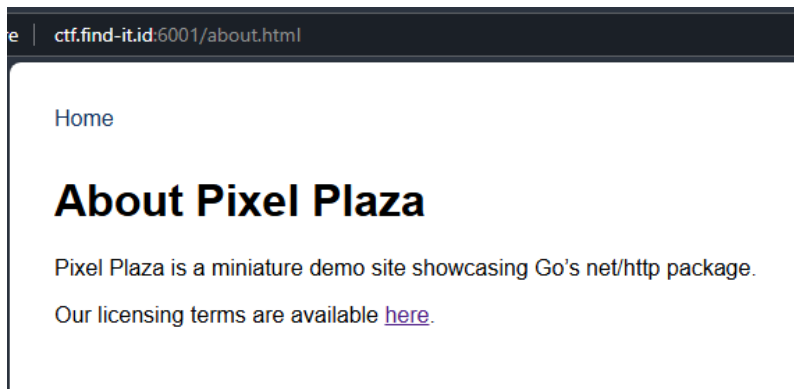
a) /gallery.html



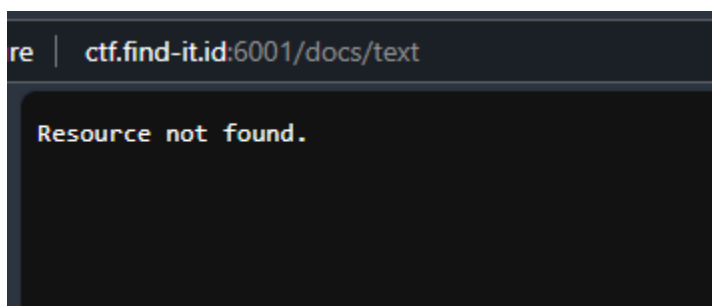
b) /contact.html



c) /about.html



d) /docs/text



4. Mungkin awalnya semua orang curiga dengan guestbook yang dapat menerima user input dan merendernya di homepage. Namun, hal tersebut tidak terlalu membantu

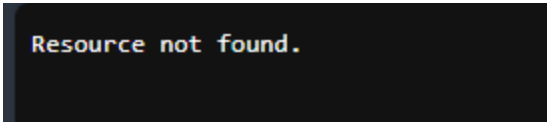
karena proses render hasil input tersebut tersanitasi dengan baik sehingga XSS tidak mungkin dilakukan.

5. Terdapat satu line yang pada kode [main.go](#) yang kami curigai, yaitu pada endpoint banner.

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

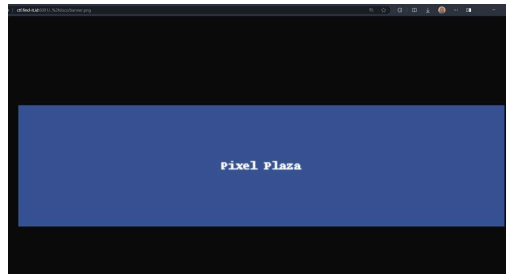
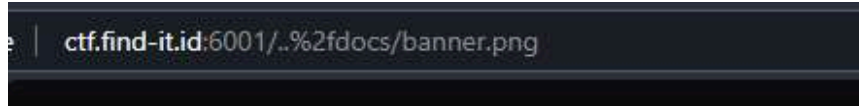
`../docs/banner.png` membuktikan bahwa file traversal mungkin saja dapat dilakukan.

6. Sayangnya, setelah kami coba membuka laman tersebut entah mengapa laman tersebut menampilkan resource not found.

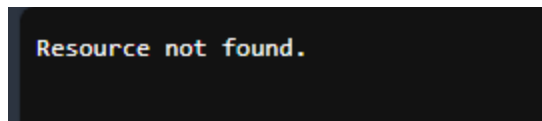


Resource not found.

7. Kami sempat menguji endpoint `/statik` tetapi fungsi `fileServer` hanya memungkinkan untuk mencari file di dalam direktori statik, di mana setelah kami coba berbagai nama secara brute force tidak dapat menemukan flag.
8. Lalu kami terpikir akan suatu hal. Bagaimana kalau sebenarnya `../docs/banner.png` itu menampilkan resource not found bukan karena tidak ada (karena memang seharusnya ada), tetapi karena terdapat suatu mekanisme yang mencegah file traversal.
9. Setelah melakukan berbagai riset, ternyata file traversal dapat dilakukan dengan mengganti char `'/'` dengan `%2f` pada url. Metode tersebut terbukti berhasil karena kami berhasil membuka `/docs/banner.png`

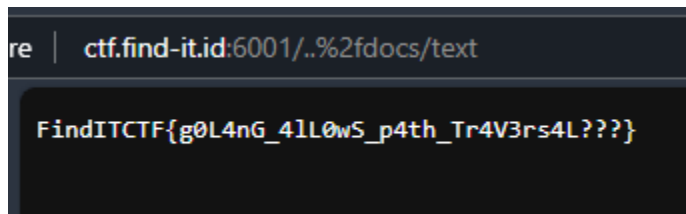


10. Namun, kami masih belum mengetahui lokasi pasti dari flag atau pun nama dari file flag. Selama setengah jam lebih kami melakukan file traversal secara brute force untuk mencoba mencari flag. Kami bahkan berhasil mendapatkan /etc/passwd.
11. Setelah beberapa waktu, terdapat satu hal yang kami terlewat, yaitu link pada laman /about.html yang mengarahkan kami ke /docs/text



Mengingat /docs/banner.png ternyata dapat dibuka, bagaimana kalau hal yang sama berlaku untuk /docs/text dan sebenarnya ada isi dari /docs/text?

12. Benar saja, setelah diuji, ternyata flag ditemukan pada path /docs/text dengan melakukan file traversal.



FindITCTF{g0L4nG_4lL0wS_p4th_Tr4V3rs4L???