

# No PWN No CRY

[ Write-up COMPFEST 15 Capture The Flag]

12 Jam nyoba PWN & krypto  
tapi masih belum ke solve be like:



**DIMANA FLAGNYA NJIRR**

Write Up By :

**PwnEater**

**Bilan**

**Flxnzz**

# Daftar Isi

<b>Forensics</b>	<b>3</b>
E2EBleed	3
Flag: COMPFEST15{tH4T5_n0T_H0w_y0u_3XchAnGe_KeYS!!}	8
Industrialspy (UNSOLVED)	8
<b>Reverse Engineering</b>	<b>10</b>
Hacked LOL	10
Flag: COMPFEST15{b1G_brr41nz_us1ng_c0d3_4s_k3y_8d7113ecc1}	11
<b>OSINT</b>	<b>12</b>
Not A CIA Test	12
Flag: COMPFEST15{Dosandaero_Gangnam_G2FW+QP}	13
<b>Misc</b>	<b>14</b>
Classroom	14
Flag: COMPFEST15{v3ry_e4sY}	14
Feedback	14
Flag:	
COMPFEST15{makasih_mas_mbak_udah_ngisi_form_tahun_depan_ikut_lagi_ya_mantap}	14
Sanity Check	14
Flag: COMPFEST15{hope_you_enjoy_the_competition_good_luck}	14

# Forensics

## E2EBleed

Diberikan sebuah file pcap dan source code dari aplikasi (frontend dan backend). Pertama analisis terlebih dahulu file pcap, dan ditemukan app untuk berkomunikasi yang menggunakan websocket :

```
Sec-WebSocket-Extensions: permessage-deflate
Sec-WebSocket-Key: GetlBu+xiI7pYmVMEcpSQ==
Connection: keep-alive, Upgrade
Pragma: no-cache
Cache-Control: no-cache
Upgrade: websocket

HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: QzMkQvXwPXXusaIXtKaoekp5tTc=

....X.#.!.d.b.h.=.u.t.e.,#.t.=.o.5.#.z.z.#.9.r.7.e.b.`.<|...j
{"type":"server","action":"auth","message":"Authenticated"}.....
.....5{"type":"server","action":"message","message":"Sent"}..~..
{"type":"init","data":
{"fromUsername":"dog","targetUsername":"cat","type":"v","value":"29116566394151601664610069303747715855356214872963782920725074996047493096331676476821431269056879517823568772
53676085562049009904952754418717197150981586225815549031392562088448468604822878613010112054104744888761117692445172587706262182471991231807304165214251814009417520153314103
76003413756444127437320157959024490402548889594399428423158315665442913934127621564548165458749280769030071246449152445642132075364463001098975402977910100272350246976959434
0720717444312470173187516371663289214255163662649108532047017008895584310143762363482235570654081243421477107783270133737738593144570052255451657875507161"}....}}...
.....Q.....G.....G.....
.....Q.....LK..JO..DN..M.....G.....NO...EH..LI..LE..DM..EE..OE..DH..KH..JL..ND..KO..JJ..NE..KL..DE..HM..JM..MK..HE..OI..KM..OH..KH..IL..JE..DE..OM..DO..KJ..JL..O
I..LJ..ML..JE..KN..JD..DH..MH..HI..MO..LM..IM..IK..OK..DL..ME..KH..EK..MN..KI..OE..HO..DJ..NE..LD..KI..KK..LK..JM..MN..DD..NH..HM..MH..NL..DO..KM..NE..NJ..NL..JI..NO..EK..IE..
ED..LD..EI..JD..NM..KJ..NM..OD..EE..OE..JL..EO..KJ..LN..DM..EE..LO..KE..NO..IK..ND..JM..JL..ED..DJ..KK..IO..MI..NM..EL..LE..MJ..LD..DN..ON..LM..DM..JD..KN..LM..NE..HI..EE..LN..
ON..EO..NO..EE..MK..JL..MI..MK..KL..LH..IM..HL..DJ..EH..KL..LE..EO..DD..EO..JO..EJ..LL..IO..HI..MO..HI..OM..HH..KI..NN..OL..OD..LI..JI..MM....
5{"type":"server","action":"message","message":"Sent"}..~..{"type":"message","data":
{"fromUsername":"dog","targetUsername":"cat","id":"1683723702544","message":"16933447801662887870119852964720377371216954236996294857522399514142220176045378344738146138733100
5488122578970145348486508894914674483621923292733602364843488016904590921800484708799926552913513027665275787380707915328348876818203061899347796374243143575017651232058140991
326091934374460892228735796441731040904332698015809803594059341707392539576924403935691864471531957264668335416830811566996896638440434775109991060789460716021882688883232346
144756615498614187013311453892051018660670528419720952463029639266645403105057124697237194038708437419288581036831468900012166367532466217182777550409137287586779946207"}...
...\\u...(.~o..9...;...~...w...1.....9w...?..pw.....
2...~o..3...~...fw..df..ob..ea..pw../...
9w..e@..mE..oD..oB..hG..kA..nM..nB..jM..eC..mM..eF..eE..jE..nD..mQ..iE..mL..kL..eF..l@..il..h@..j@..hF..jD..dE..mL..dC..mF..oL..hM..iL..eD..hF..oM..jE..lA..j@..lC..e@..dF..iF..
kD..nC..lE..jL..eM..iB..oC..lL..iM..oA..jB..nA..dA..iC..oM..oL..j@..m@..hM..jE..mD..lG..jC..mM..eF..eF..k@..hM..jB..h@..eF..iC..eG..h@..hM..dB..o@..jC..jE..oM..jF..eG..mB..kG
..jB..mC..dF..dA..nF..kG..lG..hF..dF..iA..iB..jM..iG..l@..lB..oC..kL..eA..lE..hG..jD..eA..iL..iD..lA..k@..jL..dM..iA..lE..i@..dA..iE..mF..nB..dC..iD..iM..o@..eB..k@..eM..o
A..iC..n@..mE..jG..dG..nB..oL..oE..nA..d@..iG..dG..lA..kL..oM..nE..nC..nF..dE..mE..oM..lC..lB..~...5{"type":"server","action":"message","message":"Sent"}..~..
{"type":"message","data":
{"fromUsername":"dog","targetUsername":"cat","id":"1683723717540","message":"10759128040934552042330786494370327220310465059734557898106426331483384830774920336881694650021739
1260515329878689289058342711473768915882297113276848007561180233831938676852500190042874028172031869630736728916631695131458717020636032749101807198859201271666472906342837397
773497346479056608569766047876123643505106764266752869089320021352975285455144457725916481975757236098104515056422102859597805706711622139693518169778407842508125102354843431
3448271048847767462228719784714758256760576949454573775282064370613783424874483411040327531091225486701249588418067535704762179229313716213259035758503944320096714371661"}...
.GX>=<zJD7=..e5[N49YXet.Y&_..)#.[57Sh4=LS85[.].z]\3z..39LZ",kN"P\*=...eQZet.T#z..vn..pj
un
.tz..*=MN&?[]z.
tk..sm...m
~j
.v'..rj. rn.
~'..rm ..h..sk..sk
.rm
qk...h.
```

Terdapat 2 type data yang dikirimkan, pertama **v** dan **message**, kita cari tau bagaimana app menggunakan message dan v tersebut. ditemukan potongan source code sebagai berikut pada frontend yang telah dibuild :

```
if (t.q == null || t.n == null) return z.jsx("p", {
  children: "Still exchanging keys"
});
const l = (t.p - 1n) * (t.q - 1n),
  o = bp(0x10001n, l);
return z.jsx(z.Fragment, {
  children: n.map(i => z.jsx("div", {
    className: "chat " + (i.fromUsername == r ? "chat-end" : "chat-start"),
    children: z.jsx("div", {
      className: "chat-bubble",
      children: qp(BigInt(i.message), o, t.n)
    })
  }), i.id + "-" + i.fromUsername))
})
},
rm = Ss.memo(rm)
```

Terlihat bahwa test yang akan ditampilkan di halaman chat akan proses pada fungsi `qp()` dengan parameter nilai Big Integer dari message terenskripsi, variable `o` yang berisi nilai eksponen **0x10001** dan nilai modulus **n**. Perumusan Ke 3 parameter tersebut awalnya diperoleh dari potongan kode berikut :

```
    })
  }, []), h = _.useCallback(async y => {
    if (y.type !== "init") return;
    const k = y.data.fromUsername,
        j = BigInt(y.data.value);
    if (j < 2n ** 1024n) {
      const c = j + 2n ** 1024n,
            d = await ps(1024),
            v = c * d;
      r({
        type: "init",
        data: {
          fromUsername: t,
          targetUsername: k,
          type: "v",
          value: v.toString()
        }
      }), a(S => {
        const C = {
          ...S
        };
        return C[k] = {
          p: c,
          q: d,
          n: v
        }, C
      })
    } else a(c => {
      const d = {
        ...c
      },
            v = d[k].p,
            S = BigInt(y.data.value);
      return d[k] = {
        p: v,
        n: S,
        q: S / v
      }, d
    });
  });
  u(c => {
```

Terdapat 2 kondisi, namun kita akan berfokus pada kondisi else untuk teks yang lebih panjang terlebih dulu. Implementasi ulang mekanisme tersebut dan copy juga seluruh fungsi yang berkaitan dengan prosesnya maka diperoleh kode sebagai berikut :

```

function bp(e, t) {
  for (var n = e, r = t, l = 1n, o = 0n; r > 0n;) {
    var i = n / r,
        u = r;
    r = n - i * r, n = u;
    var s = o;
    o = l - i * o, l = s
  }
  return l < 0 && (l += t), l
}

function pl(e, t, n) {
  return t == 0n ? 1n : t % 2n == 0n ? pl(e, t / 2n, n) ** 2n % n : e * pl(e, t - 1n, n) % n
}

function qp(e, t, n) {
  const r = pl(e, t, n),
        l = Zp(r);
  return new TextDecoder().decode(l)
}

function Gp(e) {
  e = e.reverse();
  let t = BigInt(0);
  for (let n = 0; n < e.length; n++) t = t * BigInt(256) + BigInt(e[n]);
  return t
}

function Zp(e) {
  let t = new Uint8Array(200),
      n = 0;
  for (; e > 0n;) t[n] = Number(e % 256n), e = e / 256n, n += 1;
  return t.subarray(0, n)
}

function Jp(e, t) {
  const n = new TextEncoder;
  let r = Gp(n.encode(e));
  return pl(r, 0x10001n, t)
}

function bp(e, t) {
  for (var n = e, r = t, l = 1n, o = 0n; r > 0n;) {
    var i = n / r,
        u = r;
    r = n - i * r, n = u;
    var s = o;
    o = l - i * o, l = s
  }
  return l < 0 && (l += t), l
}

var p = 172469508628365404723321882828991196387481476537345092348616880359100074055988026998233608818404937
var s = 291165663941516016646100693037477158553562148729637829287250749960474930963316764768214312690568795
var q = S/p //v + 2n ** 1024n
n = S
//var msg1 = 291165663941516016646100693037477158553562148729637829287250749960474930963316764768214312690568795
var msg2 = 169334478016628878701198529647203773712169542369962948575223995141422201760453783447381461387331
var msg1 = 107591280409345520423307864943703272203104650597345578981064263314833848307749203368816946500211
var msg3 = 2190829916562548777082863882356760858078848477422624231611891772542764023370463049621741036719267

var l = (p - 1n) * (q - 1n)
var o = bp(0x10001n, l);

console.log(qp(msg1, o, n))
console.log(qp(msg2, o, n))
console.log(qp(msg3, o, n))

```

Run code tersebut, namun dari pesan2 tersebut, tidak diperoleh flag :)

```

heapyarms@DESKTOP-B1PJ410:/mnt/c/Users/Heapy Arms/Downloads/Compfest/cat_dog_for/chall/dist$ node solver.js
I still don't know what you're talking about.
I don't know what you're talking about.
Alright, here are the catnips.
heapyarms@DESKTOP-B1PJ410:/mnt/c/Users/Heapy Arms/Downloads/Compfest/cat_dog_for/chall/dist$

```

Ternyata terdapat beberapa bagian paket yang di MASK, berikut contohnya :

```
192.168.1.6      TCP      54 50616 → 555 [ACK] Seq=1738 Ack=1738 Win=64050 Len=0
192.168.1.6      WebSoc... 783 WebSocket Text [FIN] [MASKED]
10.0.2.15        TCP      60 555 → 50616 [ACK] Seq=1738 Ack=2379 Win=65535 Len=0
10.0.2.15        WebSoc... 109 WebSocket Text [FIN]
192.168.1.6      TCP      54 50616 → 555 [ACK] Seq=2379 Ack=1793 Win=64050 Len=0
10.0.2.15        WebSoc... 780 WebSocket Text [FIN]
192.168.1.6      TCP      54 50616 → 555 [ACK] Seq=2379 Ack=2519 Win=64050 Len=0
192.168.1.6      WebSoc... 783 WebSocket Text [FIN] [MASKED]
10.0.2.15        TCP      60 555 → 50616 [ACK] Seq=2519 Ack=3108 Win=65535 Len=0
10.0.2.15        WebSoc... 109 WebSocket Text [FIN]
192.168.1.6      TCP      54 50616 → 555 [ACK] Seq=3108 Ack=2574 Win=64050 Len=0
10.0.2.15        WebSoc... 780 WebSocket Text [FIN]
192.168.1.6      TCP      54 50616 → 555 [ACK] Seq=3108 Ack=3300 Win=64050 Len=0
192.168.1.6      WebSoc... 784 WebSocket Text [FIN] [MASKED]
10.0.2.15        TCP      60 555 → 50616 [ACK] Seq=3300 Ack=3838 Win=65535 Len=0
10.0.2.15        WebSoc... 109 WebSocket Text [FIN]
192.168.1.6      TCP      54 50616 → 555 [ACK] Seq=3838 Ack=3355 Win=64050 Len=0
192.168.1.6      WebSoc... 62 WebSocket Connection Close [FIN] [MASKED]
```

Untuk mendapatkan key untuk xor data yg di mask sangat mudah, xor saja dengan key yang terdapat di paket data tersebut :

```
▼ WebSocket
  1... .. = Fin: True
  .000 ... = Reserved: 0x0
  .... 1000 = Opcode: Connection Close (8)
  1... .. = Mask: True
  .000 0010 = Payload length: 2
  Masking-Key: a0d37a57
  Masked payload
  > Payload
```



Gunakan script berikut untuk unmasking :

```
msg": "\x3c\x7a\x4a\x44\x37\x3d\x1c\x07\x65\x35\x5b\x4e\x34\x39\x59\x58" \
"\x65\x74\x1c\x59\x26\x2c\x5f\x1f\x7d\x23\x1c\x5b\x35\x37\x53\x68" \
"\x34\x3d\x4c\x53\x26\x35\x5b\x1f\x7d\x7a\x5d\x5c\x33\x7a\x12\x1f" \
"\x33\x39\x4c\x5a\x22\x2c\x6b\x4e\x22\x2a\x50\x5c\x2a\x3d\x1c\x07" \
"\x65\x3c\x51\x5a\x65\x74\x1c\x54\x23\x7a\x04\x1f\x76\x6e\x06\x0e" \
"\x70\x6a\x0d\x0a\x75\x6e\x0d\x08\x74\x7a\x12\x1f\x2a\x3d\x4d\x4e" \
"\x26\x3f\x5b\x1f\x7d\x7a\x0b\x0d\x74\x6b\x0e\x04\x73\x6d\x0c\x0e" \
"\x7f\x6d\x0d\x0a\x7e\x6a\x0d\x0c\x76\x60\x0b\x0f\x72\x6a\x07\x09" \
"\x72\x6e\x0f\x0a\x7e\x60\x0f\x0c\x72\x6d\x09\x05\x7f\x68\x0b\x0f" \
"\x73\x6b\x07\x0e\x73\x6b\x0a\x04\x72\x6d\x0a\x0a\x71\x6b\x06\x0e" \
"\x7f\x68\x06\x0d\x73\x6d\x0e\x0c\x75\x6b\x08\x0e\x71\x6f\x0b\x0d" \
"\x75\x60\x0e\x05\x72\x60\x0f\x05\x70\x6c\x06\x09\x77\x6f\x07\x05" \
"\x75\x6b\x07\x09\x71\x6f\x07\x0d\x7e\x6f\x09\x0e\x70\x6f\x0c\x0e" \
"\x72\x69\x0e\x04\x70\x6c\x0b\x05\x70\x6b\x0a\x0d\x74\x6f\x08\x09" \
"\x75\x6f\x06\x0f\x73\x6c\x06\x0e\x77\x60\x09\x04\x7e\x6a\x06\x0d" \
"\x70\x6c\x0b\x0d\x76\x6a\x0e\x05\x7e\x6a\x08\x0a\x70\x61\x0d\x0e" \
"\x70\x68\x07\x0f\x7e\x6f\x0a\x0a\x74\x68\x07\x0c\x73\x69\x0e\x05" \
"\x75\x6e\x0b\x0e\x72\x60\x0f\x0d\x7e\x6c\x0b\x0c\x73\x6f\x0e\x0c" \
"\x76\x68\x0e\x08\x73\x68\x07\x0b\x71\x6b\x0b\x08\x75\x69\x0c\x08" \
"\x7e\x6d\x09\x0b\x73\x6a\x0d\x0b\x75\x68\x06\x0f\x70\x6a\x0c\x0f" \
"\x76\x6d\x0b\x0e\x73\x69\x08\x0e\x70\x6a\x06\x0a\x70\x6e\x0d\x09" \
"\x77\x68\x0b\x09\x77\x68\x0e\x0e\x74\x60\x0d\x0f\x71\x6d\x0e\x05" \
"\x7e\x6a\x0a\x0c\x7f\x69\x0a\x0d\x72\x60\x0f\x0e\x73\x69\x06\x0c" \
"\x7e\x6f\x0a\x0d\x72\x68\x09\x04\x70\x6f\x0d\x0b\x7f\x6c\x0d\x0c" \
"\x7e\x60\x06\x0d\x72\x6a\x0f\x09\x7e\x6a\x0a\x05\x72\x69\x0b\x0b" \
"\x76\x68\x0a\x04\x73\x68\x08\x09\x75\x6c\x0c\x0f\x72\x69\x09\x0f" \
"\x76\x6c\x09\x08\x73\x6e\x0f\x0c\x72\x6c\x09\x08\x70\x6b\x0c\x0f" \
"\x74\x68\x0e\x08\x76\x6f\x06\x0e\x72\x6a\x0b\x0f\x72\x6a\x07\x04" \
"\x73\x68\x0d\x0e\x72\x60\x0c\x0b\x74\x69\x0d\x0e\x71\x60\x07\x04" \
"\x7f\x6e\x0b\x0a\x7e\x69\x0d\x0e\x71\x6c\x09\x0e\x74\x60\x07\x04" \
"\x72\x60\x08\x0e\x72\x6a\x0a\x0a\x77\x68\x0d\x0b\x74\x6c\x09\x0d" \
"\x7f\x68\x07\x08\x7e\x69\x07\x08\x75\x6a\x0d\x0a\x7e\x6f\x07\x08" \
"\x7f\x6b\x07\x09\x76\x6a\x0f\x0f\x7e\x6d\x0e\x0b\x77\x60\x08\x09" \
"\x73\x6b\x07\x0c\x74\x68\x08\x0f\x70\x6e\x0f\x0c\x73\x6f\x0b\x0b" \
"\x77\x6c\x0b\x0b\x7e\x6f\x0e\x0a\x75\x6b\x08\x0c\x75\x68\x0d\x05" \
"\x7f\x60\x0b\x0c\x75\x68\x0e\x0c\x7f\x6b\x09\x05\x7e\x6d\x0f\x09" \
"\x74\x6e\x07\x04\x75\x6f\x0c\x04\x73\x6f\x08\x0a\x73\x6c\x08\x0d" \
"\x7e\x68\x07\x08\x7e\x61\x0e\x08\x77\x6f\x0c\x0a\x7f\x61\x08\x05" \
"\x76\x6a\x0e\x0b\x7f\x61\x07\x05\x73\x61\x08\x0c\x70\x6a\x07\x0a" \
"\x75\x6c\x0a\x05\x77\x6c\x0d\x0f\x7e\x69\x0d\x0e\x77\x6a\x08\x05" \
"\x75\x6e\x09\x08\x7e\x61\x0a\x0f\x74\x61\x09\x05\x72\x68\x07\x0d" \
"\x70\x61\x08\x0a\x76\x60\x0a\x08\x74\x61\x0d\x0f\x75\x6d\x06\x08" \
"\x72\x69\x06\x0c\x73\x6b\x0a\x0a\x76\x6c\x06\x08\x75\x6e\x0b\x0a" \
"\x73\x6e\x0b\x0e\x75\x69\x0f\x0a\x71\x6d\x07\x08\x7e\x6c\x0b\x0a" \
"\x71\x68\x07\x08\x71\x6f\x0e\x08\x7f\x6f\x07\x0f\x72\x61\x1c\x40\x3a"
, "key": "\x47\x58\x3e\x3d"
# {msg:,key:}
# {msg:,key:}
# {msg:,key:}
}
for c in data:
    unmasked_1 = ""
    for x in range(len(c["msg"])):
        unmasked_1 += chr(ord(c["msg"][x]) ^ ord(c["key"][x%(len(c["key"])])))
    print(unmasked_1)
```

Run script tersebut dan diperoleh plain text nya :

```
heapyarns@DESKTOP-B1PJ418: /mnt/c/Users/Heapy_Arws/Downloads/Compfest/cat_dog_for/chall/dist$ python3 unmask.py
{"type": "ident", "data": {"username": "cat", "password": "asd"}}
{"type": "init", "data": {"fromUsername": "cat", "targetUsername": "dog", "type": "v", "value": "-729980485786618604968636249911276974316221356885564924813200798632601749512936134474868594002598110
16841791728908184368808047274958302035161692729233567955186286938221663486655209472701043476532141631154911363114070383390894275218903520691740151755828731018277983381985455158526945931
6196217467998181217"}}
{"type": "message", "data": {"fromUsername": "cat", "targetUsername": "dog", "id": "1683723693610", "message": "3632788507148418529006889428869509656171243977620776238366180902350677060064558192432
6034251065464154782698382049206673371372405174301817890630379259560058154980219102840804468260891870851655086710376648028635295977838211968649366271647704803259995356450468574318092206072
38753747314670451532828614488389751974847170773805667953051290988722850712582246751139490808826127868433200466839197032712989999728666342220434305681721841079919649314232110499024795063671063
384854138007137523382773268890066971990443063061351522401951389799857061091887826099368296725487981139426754408020154712082554264793361214829491439745400"}}
{"type": "message", "data": {"fromUsername": "cat", "targetUsername": "dog", "id": "1683723707949", "message": "95991081315237784213743628912735686496181840936890846079213215195038199879199338054659
83451365024391613480401981869613083940483359139198432038816054049265286685959383245340714626780022696898655742363609825802343767022416846356323862393565961553488760191186025066541840931393
687512486667945809319567692404504481787663521660160283863637692311766727067841612834284118421233872670231431483035481579868205244055107953665791594970011427761429477562514404157543697288
355487089757284595061136227828668517158233557971175799810347656002567102262258293270039883087246585945232821604527939387020762619239180451022384706110752"}}
{"type": "message", "data": {"fromUsername": "cat", "targetUsername": "dog", "id": "1683723726353", "message": "503309452385370231185259456170811557808549393430956763830800450123636750280858107404
07063394679097773723510971050724037642782408308799280745012089267793709297473091410826535810945147011085409663552125057642362082722215534163782776340054000338326508024181405813418197405079
733684319880521492485156104940464242251721475461154757322308517835252594083358263136899865791336473389958635247083634708095919522379795839412129506086443913862761147560456970776212038851
2081837895143699272947674469095990507278968120689984961729724480432913302682675994239785090796718453932258551814347414852657465321176595945768956505879259"}}
heapyarns@DESKTOP-B1PJ418: /mnt/c/Users/Heapy_Arws/Downloads/Compfest/cat_dog_for/chall/dist$
```

Masukan cipher cipher tersebut ke solver script awal tadi maka diperoleh flag sebagai berikut :

```
heapyarms@DESKTOP-B1PJ410:/mnt/c/Users/Heapy_Arms/Downloads/Compfest/cat_dog_for/chall/dist$ node solver.js
I still don't know what you're talking about.
I don't know what you're talking about.
Alright, here are the catnips.
Heyo, you got the stuffs?
Don't be annoying, you know it's me.
Tch, COMPFEST15{tH4T5_n0T_H0w_y0u_3XchAnGe_KeYS!!}
```

**Flag: COMPFEST15{tH4T5\_n0T\_H0w\_y0u\_3XchAnGe\_KeYS!!}**

## Industrialspy (UNSOLVED)

Diberikan sebuah memory dump, lakukan analisis dengan volatility dengan profil Win7SP1x64  
Diberikan sebuah memory dump, lakukan analisis dengan volatility dengan profil Win7SP1x64.  
Lihat process list memory tersebut dengan pslist. Berdasarkan deskripsi, "I have suspicions that our graphic designer intern" maka target kita yaitu menemukan software yang biasa digunakan graphic designer. Lalu saya menemukan process dengan PID 1320, yaitu mspaint.exe.

0xfffffa8003de21e0	SearchIndexer.	1932	520	15	546	0	0	2023-07-12 06:58:16 UTC+0000
0xfffffa8003e73b30	mspaint.exe	1320	1628	8	161	1	0	2023-07-12 06:58:26 UTC+0000
0xfffffa8003e8e390	svchost.exe	1460	520	9	110	0	0	2023-07-12 06:58:26 UTC+0000

Extract PID tersebut dengan **volatility -f lyubov\_20230712.mem --profile=Win7SP1x64 memdump -p 1320 -D .**

```
flxnzz@Wzrd:~/Downloads/Compfest-Quals$ volatility -f lyubov_20230712.mem --profile=Win7SP1x64 memdump -p 1320 -D .
Volatility Foundation Volatility Framework 2.6
*****
Writing mspaint.exe [ 1320] to 1320.dmp
```

Maka akan terdapat file baru bernama 1320.raw. Saya mengganti formatnya dengan .data agar bisa dibuka lewat GIMP

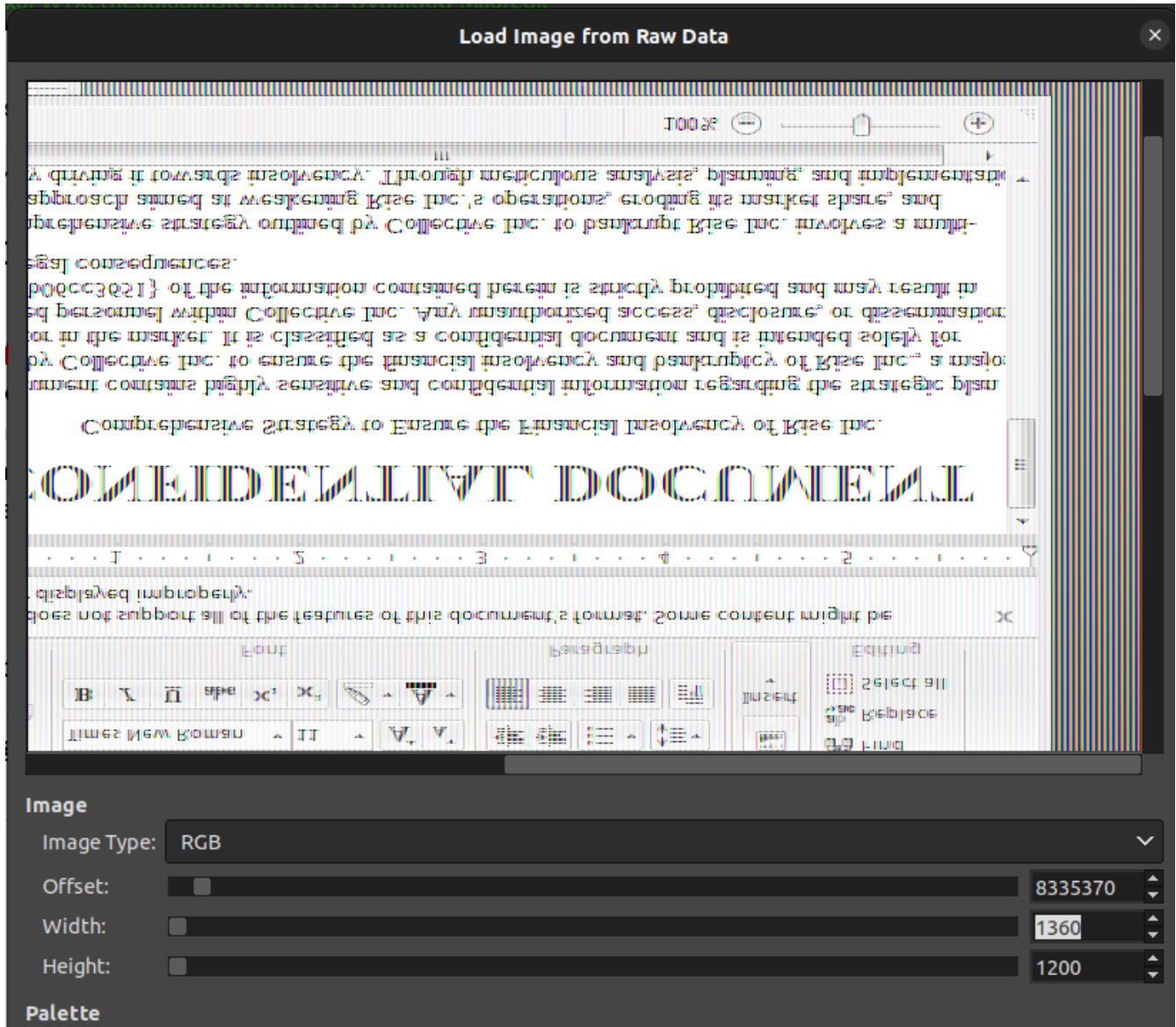
Buka file .data tadi via GIMP dan masukan value :

Offset : 8335370

Width : 1360

Height : 1000





Flag perlu dibalik-balik agar bisa terbaca.

Diperoleh source code yang di strip, setelah ditelaah, pada lingkaran merah, key didapat dengan membaca source code `__file__` yang merupakan file `helper.py`. Lingkaran kuning merupakan bagian iterasi setiap file yang bukan merupakan file `.py` dan pada lingkaran hijau, setiap byte data pada sebuah file di iterasi dan di xor dengan operasi `file[x] ^ key[x%len(key)]`

Dibuatlah sebuah solver sebagai berikut :

```
key = open('helper.py', 'r').read()
data = open('important_file.hackedlol', 'r').read()
flag = ""
for x in range(len(data)):
    flag += chr(ord(data[x]) ^ ord(key[(x*0x27)%len(key)]))
print(flag)
```

```
The flag is: COMPFEST15{b1G_brr4lnz_us1ng_c0d3_4s_k3y_8d7113ecc1}
```

Flag: COMPFEST15{b1G\_brr4lnz\_us1ng\_c0d3\_4s\_k3y\_8d7113ecc1}

# OSINT

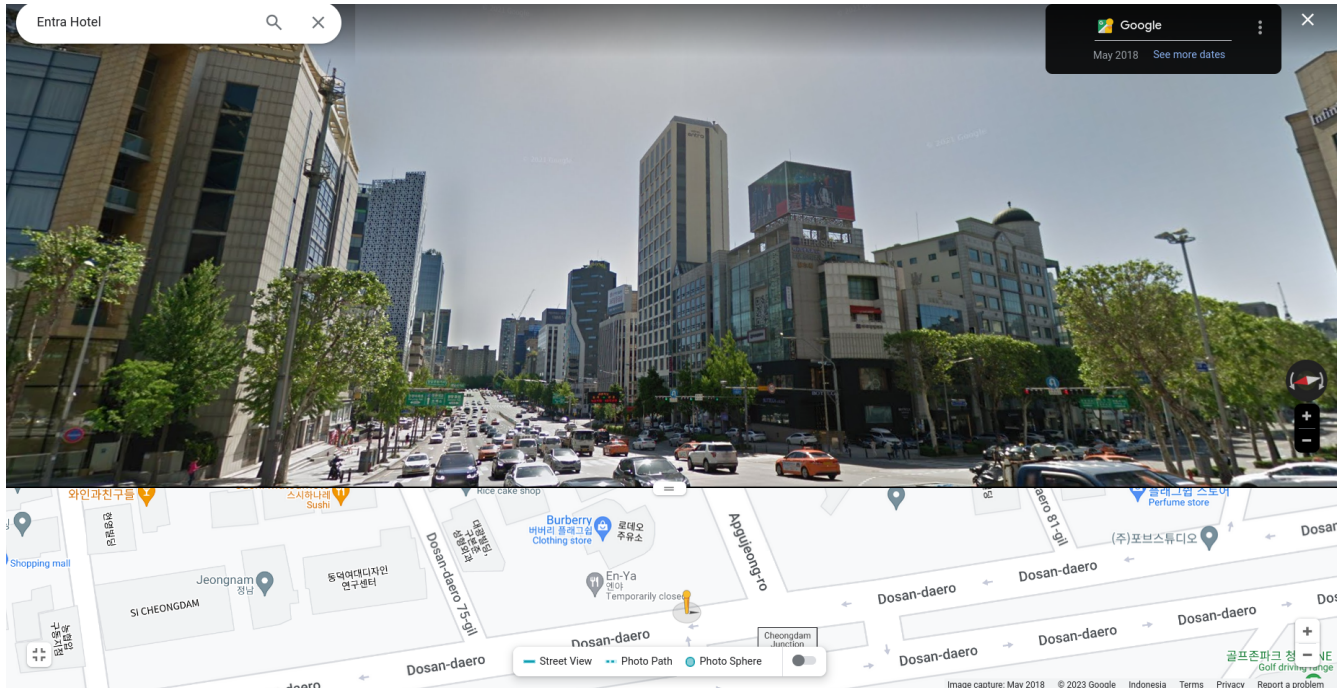
## Not A CIA Test

Diberikan sebuah file gambar yang diambil dari Instagram seorang artis Korea bernama An Yujin. Berdasarkan deskripsi soal, foto ini diambil di dekat Burberry Store di Seoul, sedangkan alamat pastinya adalah yang perlu kita temukan.

Karena gambar ini cukup burik, jadi saya tingkatkan kualitas gambarnya dengan bantuan tools online, VanceAI Image Enhancer. Lalu akan terlihat tulisan di rambu penunjuk jalan terbaca samar-samar.



Untuk memastikannya tinggal search di google, lalu didapatkan Jammon Hangang Park. Setelah itu tinggal cari Burberry Store sekitar wilayah tsb dengan google maps.



Setelah mencari2, saya menemukan Lokasi yang tepat seperti pada gambar challenge, yaitu di dekat [Hotel Entra Gangnam](#).

Alamat Burberry adalah: 459 **Dosan-daero, Gangnam-gu, Seoul, South Korea**.

Plus code Burberry: **G2FW+QP** Seoul, South Korea

Jadi flagnya kurang lebih seperti ini (*saya lupa yg di sambit seperti apa*)

**Flag: COMPFEST15{Dosandaero\_Gangnam\_G2FW+QP}**

# Misc

## Classroom

Pada spreadsheet terdapat string base 64 yang jika didecode maka akan muncul pesan:

Aku menyembunyikan flagnya di jadwal Hari Selasa karena kukira tidak ada murid yang secerdas itu!

Row 8 pada spreadsheet merupakan nilai yang akan kita cari di kolom dan baris pada sheet Flag.

Flag: `COMPFEST15{v3ry_e4sY}`

## Feedback

### Feedback Penyisihan CTF COMPFEST 15

Terima kasih!

`COMPFEST15{makasih_mas_mbak_udah_ngisi_form_tahun_depan_ikut_lagi_ya_mantap}`

[Submit another response](#)

Flag:

`COMPFEST15{makasih_mas_mbak_udah_ngisi_form_tahun_depan_ikut_lagi_ya_mantap}`

## Sanity Check

Diberikan flag, dan makasih 🙏

### Welcome to #first-blood!

This is the start of the #first-blood channel. `COMPFEST15{hope_you_enjoy_the_competition_good_luck}`

Flag: `COMPFEST15{hope_you_enjoy_the_competition_good_luck}`