No PWN No CRY

[Write-up COMPFEST 15 Capture The Flag]

Write Up By :

PwnEater

Bilan

Flxnzz

Daftar Isi

Forensics **3**

E2EBleed 3

Flag: COMPFEST15{tH4T5_n0T_H0w_y0u_3XchAnGe_KeYS!!} 8

Industrialspy (UNSOLVED) 8

Reverse Engineering **10**

Hacked LOL 10

Flag: COMPFEST15{b1G_brr41nz_us1ng_c0d3_4s_k3y_8d7113ecc1} 11

OSINT **12**

Not A CIA Test 12

Flag: COMPFEST15{Dosandaero_Gangnam_G2FW+QP} 13

Misc **14**

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 :

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))
    })
},
```

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 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 = 291165663941516016646100693037477158553562148729637829207250749960474930963316764768214312690568795
var q =  S/p //v + 2n ** 1024n
n = S
//var msg1 = 291165663941516016646100693037477158553562148729637829207250749960474930963316764768214312690
var msg2 = 169334478016628878701198529647203773712169542369962948575223995141422201760453783447381461387331
var msg1 = 107591280409345528423307864943703272203104650597345578981064263314833848307749203368816946500211
var msg3 = 219082991656254877702863882356760858078848477422624231611891772542764023370463049621741036719267

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 kode 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=2050 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\x6d\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 :

```
heapyarms@DESKTOP-B1PJ410: /mnt/c/Users/Heapy Arms/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":"~7299804857866186049608636249911276974316221356885564924813200798632601749512936134947486850400025981101684179172890018430688080472749583020351616927922933567955186286938222166348665520947270104347653214163115491136311407038339089427521890352069174015175582873101827798330198545515852694593181962174679981812l7"}}
{"type":"message","data":{"fromUsername":"cat","targetUsername":"dog","id":"1683723693610","message":"363278850714841852900688942886950965617124397762077762383661809823506770640645581924326034251065464154782698382024920667337137240517430181789063037925956005815498021910284084468626089187085165508671037664802863529597783821196864936627164770480325999535645004605743180922060723875374731467411532828614488838975197484717977305667953051290988722850712582246751139490068826127868433200466839197032712989999972866634222043430568172184107991964931423211049902479506367116633848543188071375233382773256889006669711990443063061351522401951389799857061091887826099368296725487981139426754400201547120825546247933612148294914397454001"}}
{"type":"message","data":{"fromUsername":"cat","targetUsername":"dog","id":"1683723707949","message":"959910813152377842137436289127356864961818409368908460792132151950381998791993380546598345136502439161348040198186961308394048335913919843203881605404926528060595938324534071462678002269689865574236360982580234376702241684635632386239356596155348876019118602506654184409313936875124866679845809319567692404504481787663521660160283863637692311766727067841612834284118421233872670231431483035481579868205244055107953665791594970011427761429477596251440415754369728835548700975572845950611362278286685171582335579711757998103476560025671022622582932700398830872465859452328216045279393870207626192391804510223804706110752"}}
{"type":"message","data":{"fromUsername":"cat","targetUsername":"dog","id":"1683723726353","message":"503389452385379231185252945617981155788652439343495547638380804501236367502808581874840798239467909773772351097458734037642782448430879928074501208926779337092974730914108265358109451470110054096635521259576423620827222155341637287763400540003383265089241814058134181974050797736884319880521492485156104940464242251721475461154757322300517835252529940335826313368998657913364733899586352470036347080959919522379795839412129506086449391306276114756045697072361203888512001837895143699272947674460909599050727896812068998496172972448043291330268267599423978509079671845393225855181434714852657465321176595945760956505879259"}}
heapyarms@DESKTOP-B1PJ410: /mnt/c/Users/Heapy Arms/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.

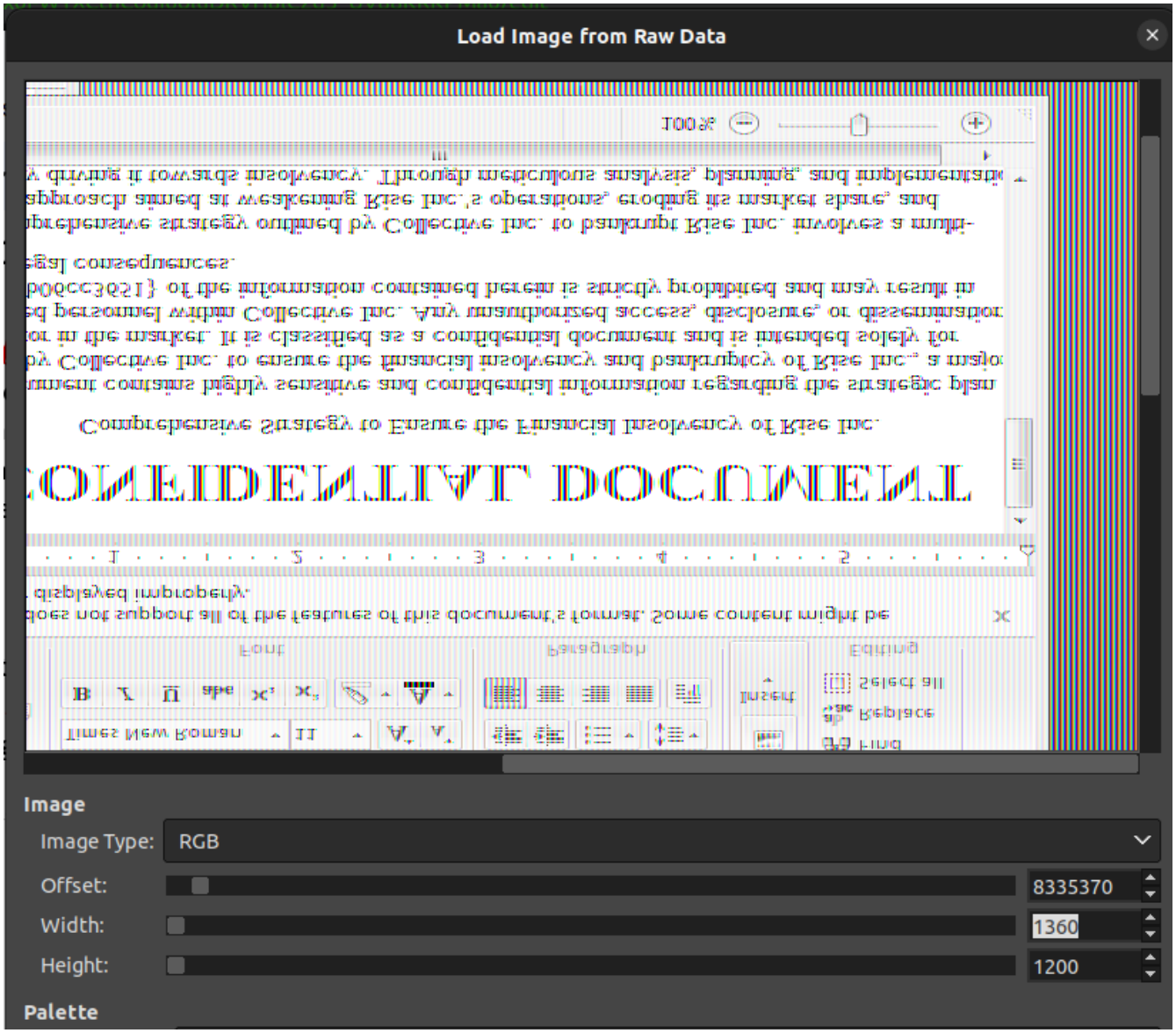
```
0xfffffa8003de21e0 SearchIndexer.      1932    520     15      546      0      0 2023-07-12 06:58:16 UTC+0000
0xfffffa8003e73b30 mspaint.exe         1320   1628      8      161      1      0 2023-07-12 06:58:26 UTC+0000
0xfffffa8003e8e390 svchost.exe         1460    520      9      110      0   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

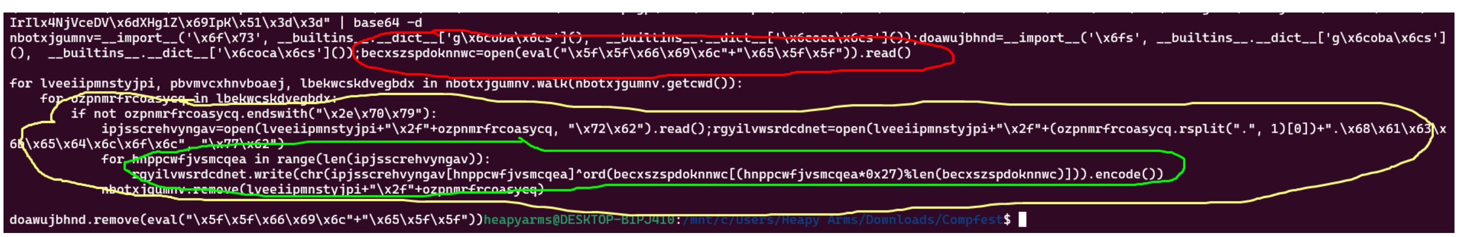
Reverse Engineering

Hacked LOL

Diberikan sebuah file pyc , gunakan uncompyle6 untuk mendekompilasi file tersebut dan diperoleh :

Decode base64 nya

Pada bagian baris terakhir terlihat, terdapat operasi I/O pada file yang merupakan file helper.py yang isinya diperoleh dari hasil decode. Sebagian data di ubah kedalam bentuk escaped string, kita decode ke base64 menggunakan echo -ne saja dan hasilnya merupakan isi dari script helper.py



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_brr41nz_us1ng_c0d3_4s_k3y_8d7113ecc1}
```

Flag: COMPFEST15{b1G_brr41nz_us1ng_c0d3_4s_k3y_8d7113ecc1}

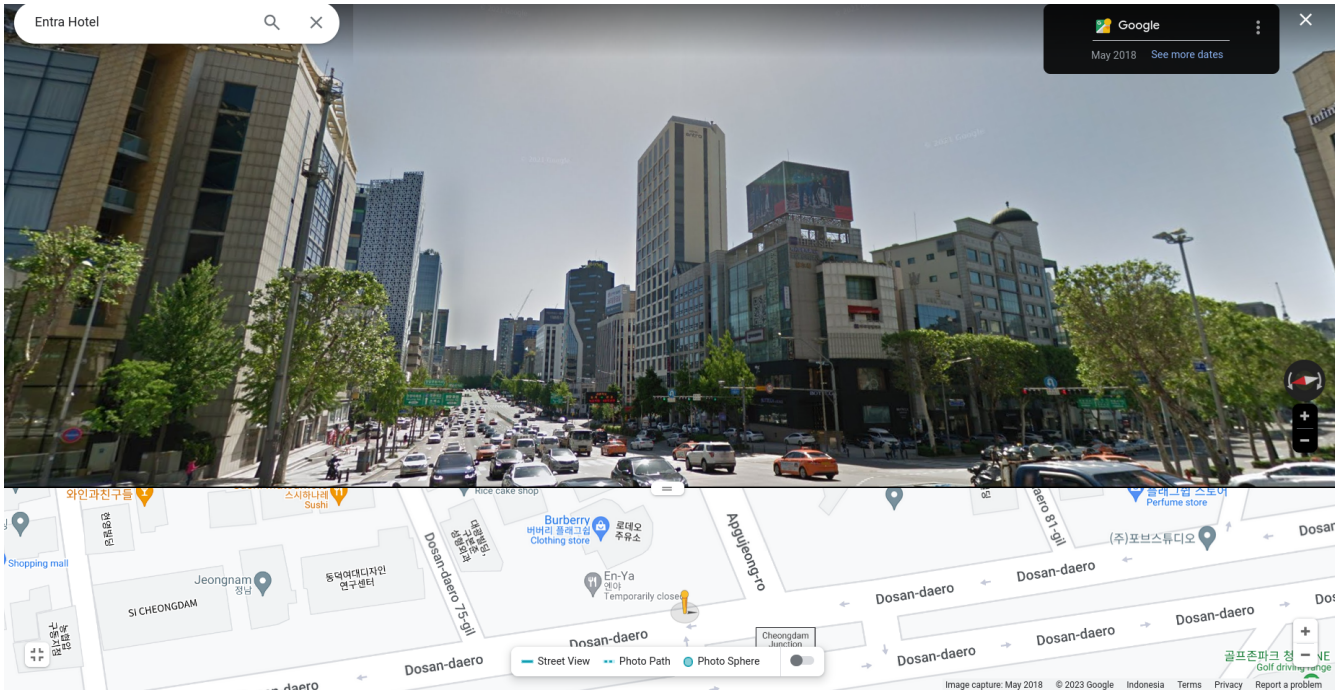
OSINT

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.

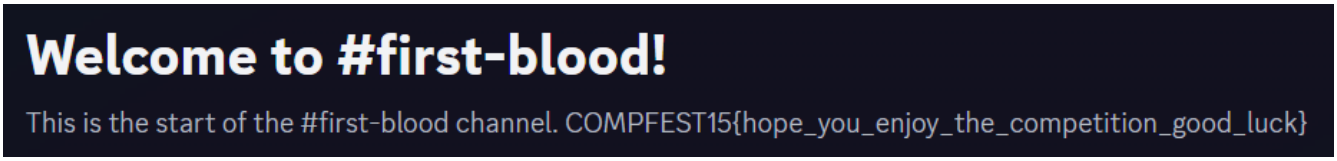
Flag: COMPFEST15{v3ry_e4sY}

Feedback

Flag:
COMPFEST15{makasih_mas_mbak_udah_ngisi_form_tahun_depan_ikut_lagi_ya_mantap}

Sanity Check

Diberikan flag, dan makasih 🙏



Flag: COMPFEST15{hope_you_enjoy_the_competition_good_luck}