

LKSMK2023 MALANG

TEAM 2 SMKN 4 MALANG :

- Firda Gheitsa Sahira
- Fikri Muhammad Abdillah

BAB A: Jeopardy

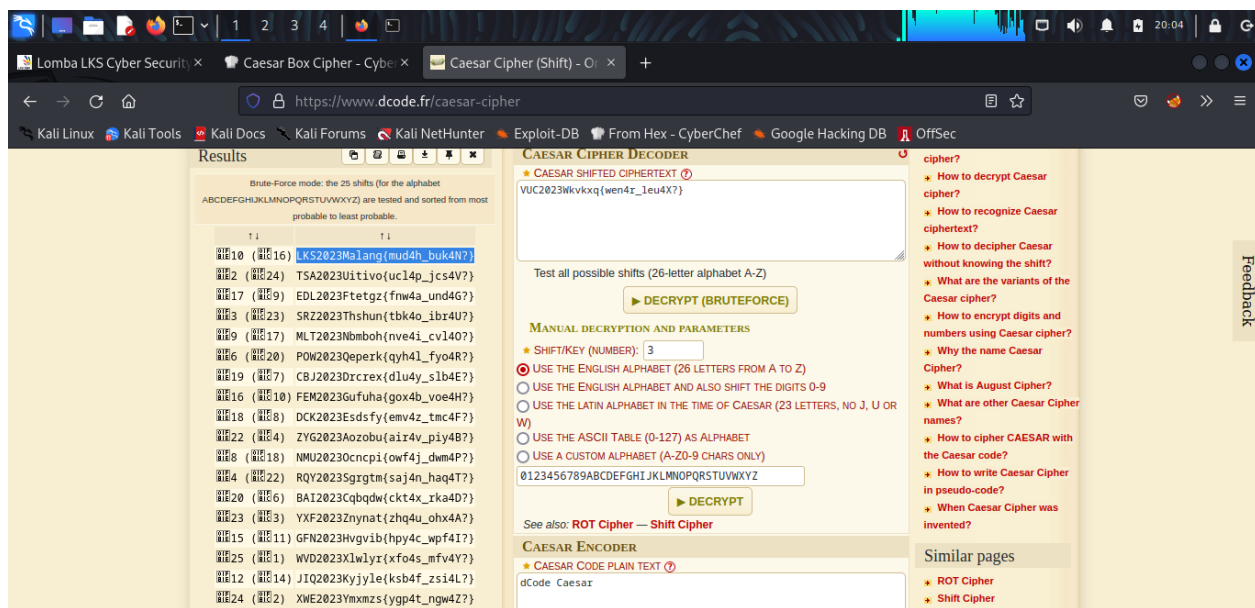
1. EasyMe (Cryptography)

isi soalnya yaitu *VUC2023Wkvkxq{wen4r_leu4X}*

setelah kita analyze metode yang digunakan adalah caesar cipher

setelah itu kita coba decode dengan tool online untuk decrypt dan terlihat flag disalah satu hasil dari output decrypt tersebut

FLAG : LKS2023Malang{mud4h_buk4N?}



2. EncDec (Cryptography)

Pertama-tama kita mendapatkan 2 file yaitu file enc.py dan encrypted.txt

enc.py berisi function encrypt dan encrypted.txt berisi FLAG yang sudah di encrypt

cara decrypt saya yaitu:

membuat loop untuk setiap karakter pada encrypted.txt, lalu program akan menambahkan counter setiap bertemu karakter "X", lalu jika program menemukan karakter "-" maka program akan mengurangi counter dengan 9 lalu di xor 0x50 dan hasil akan dimasukkan ke variabel uncipher, dan akan terus begitu sampai huruf

habis, lalu hasil algoritma tersebut akan di reverse untuk hurufnya, lalu akan mendapatkan flag:

FLAG: LKS2023Malang{tHanKs_f0r_f1nd1ng_m3}

[illegible]

3. x0r (Cryptography)

pertama-tama kami mendapatkan dua file yaitu xor.py dan xor_cipher.txt

untuk algoritmanya sama dengan waktu di encrypt, tapi untuk decrypt awalnya di decode base64 terlebih dahulu, baru memakai algoritma encrypt nya lalu akan menghasilkan flag:

FLAG: LKSMK2023{y0u_d3crypt_m3}

The screenshot shows a Kali Linux terminal window with a dark background. At the top, there's a taskbar with various application icons. The terminal title bar reads "root@kali: ~kali". Below the title bar, there are two tabs: "root@kali: ~kali" and "kali@kali: ~". The active tab is "kali@kali: ~". The terminal content shows a Python script named "crypto3.py" being edited with "GNU nano 7.2". The script defines two functions: "encrypt(plain)" and "decrypt(cipher)". The "encrypt" function takes a string "plain" and a key "n0k3y", iterates over each character, and XORs it with the corresponding character in the key (repeating the key if necessary). The result is encoded as a base64 string. The "decrypt" function takes a base64-encoded string "cipher", decodes it, and iterates over each character, XORing it with the corresponding character in the key to retrieve the original "uncipher". At the bottom, the script prints the result of decrypting a specific base64-encoded string.

```

root@kali: ~kali
kali@kali: ~
GNU nano 7.2 crypto3.py
from base64 import b64encode, b64decode

def encrypt(plain):
    key = "n0k3y"
    cipher = ""
    for c, i in enumerate(plain):
        cipher += chr(ord(i) ^ ord(key[c % 5]))
    print(b64encode(cipher.encode()))

if __name__ == "__main__":
    plain = input()
    encrypt(plain)

def decrypt(cipher):
    cipher = b64decode(cipher.encode()).decode()
    key = "n0k3y"
    uncipher = ""
    for c, i in enumerate(cipher):
        uncipher += chr(ord(i) ^ ord(key[c % 5]))

    return uncipher

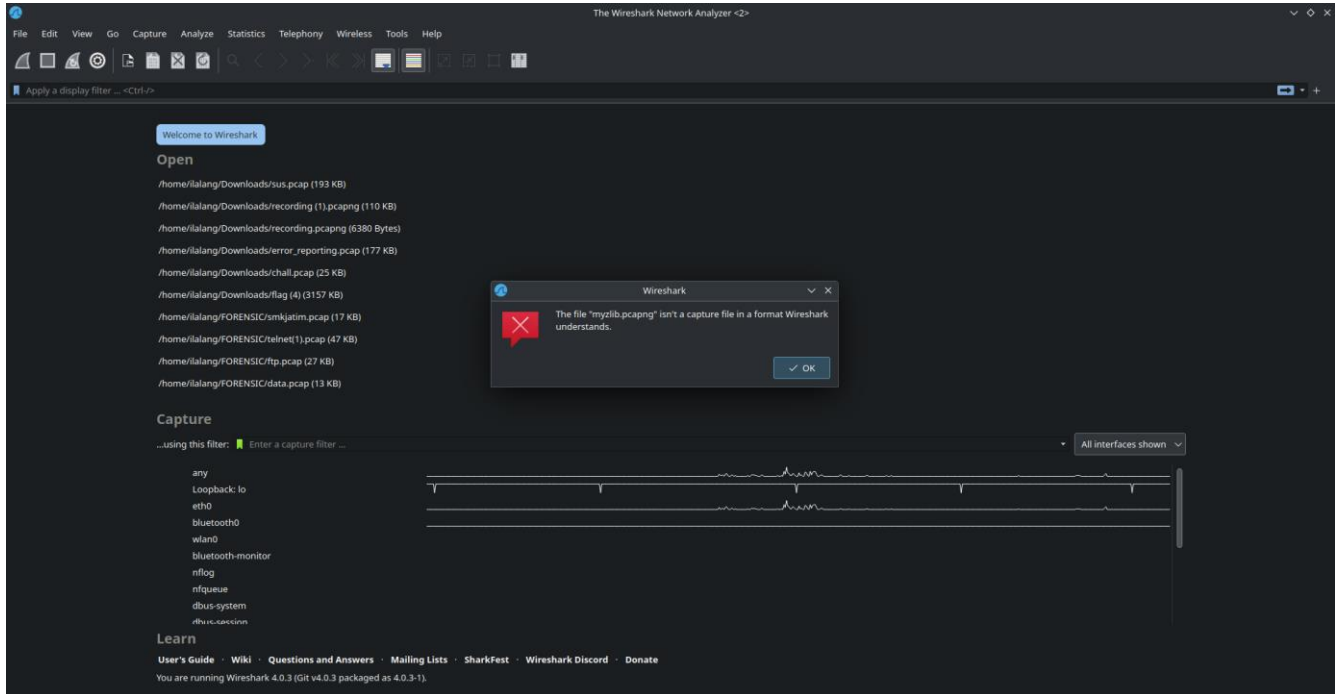
print(decrypt("Ins4AULcAyZSFQ9eDEgAIUU0V0oNQhJDDTFdWE4="))

```

4. Track

isi soalnya yaitu file myzlib.pcapng

Saya coba buka dengan wireshark tetapi gagal



saya coba analyze lagi dengan command file <file> ternyata adalah zlib

```
(root@msi)-[/home/ilalang/Downloads]
# file myzlib.pcapng
myzlib.pcapng: zlib compressed data

(root@msi)-[/home/ilalang/Downloads]
#
```

jalankan perintah `binwalk --dd='.*' myzlib.pcapng`

```
ilalang@msi:~/Downloads$ binwalk --dd='.*' myzlib.pcapng
/usr/lib/python3/dist-packages/llvmlite/llvmpy/_init_.py:3: UserWarning: The module 'llvmlite.llvmpy' is deprecated and will be removed in the future.
warnings.warn(
/usr/lib/python3/dist-packages/llvmlite/llvmpy/core.py:8: UserWarning: The module 'llvmlite.llvmpy.core' is deprecated and will be removed in the future. Equivalent functionality is provided by 'llvmlite.ir'.
warnings.warn(
/usr/lib/python3/dist-packages/llvmlite/llvmpy/passes.py:17: UserWarning: The module 'llvmlite.llvmpy.passes' is deprecated and will be removed in the future. If you are using this code, it should be inlined into your own project.
warnings.warn(

DECIMAL      HEXADECEMAL  DESCRIPTION
-----
0            0x0         Zlib compressed data, default compression

ilalang@msi:~/Downloads$
```

ada flag diantara file yang diekstrak yaitu

LKS2023Malang{zlib_c0mpr3ss1ion_0ver_pcap}

5. I am on diet (web)

kita mendapatkan file diet.webp

Saya coba dengan exiftool dan decode strings yang ada di software

setelah kita analyze ternyata menggunakan ascii

lalu kita dapat flag nya melalui tool online

LKS2023Malang{y4h_k3t4huan_d3h}

```
# file diet.webp
diet.webp: RIFF (little-endian) data, Web/P image

(root@msi)-[/home/ilalang/Downloads]
#
```

```
U9v}4
+fdGzi
R(7U6
EXIF
ExifMeta
4c4b53323032334d616c616e677b7934685f6b3374346875616e5f6433687d

(root@msi)-[/home/ilalang/Downloads]
# exiftool diet.webp
ExifTool Version Number      : 12.55
File Name                    : diet.webp
Directory                    : .
File Size                    : 104 kB
File Modification Date/Time   : 2023:02:15 08:07:28+07:00
File Access Date/Time        : 2023:02:15 08:20:55+07:00
File Inode Change Date/Time   : 2023:02:15 08:07:28+07:00
File Permissions              : -rw-r--r--
File Type                    : Extended WEBP
File Type Extension          : webp
MIME Type                    : image/webp
WebP Flags                   : EXIF
Image Width                  : 1500
Image Height                 : 1000
VP8 Version                  : 0 (bicubic reconstruction, normal loop)
Horizontal Scale              : 0
Vertical Scale                : 0
Exif Byte Order               : Big-endian (Motorola, MM)
X Resolution                  : 72
Y Resolution                  : 72
Resolution Unit               : inches
Software                     : 4c4b53323032334d616c616e677b7934685f6b3374346875616e5f6433687d
Y Cb Cr Positioning           : Centered
Image Size                   : 1500x1000
Megapixels                   : 1.5
```

The screenshot shows a web browser with an online ASCII decoder tool. The tool has a search bar at the top with the text "Search for a tool". Below the search bar, there is a section titled "ASCII CODE" with a sub-header "Informatics - Character Encoding - ASCII Code". The main content area displays the decoded ASCII string: "LKS2023Malang{y4h_k3t4huan_d3h}". The string is shown in a monospaced font, with each character represented by a small icon. The tool also includes a "Summary" section on the right with links to "ASCII Converter", "ASCII Encoder", and "What is the ASCII standard?". At the bottom, there is a "Forum/Help" section with a "DISCORD" link and a "Keywords" section.

6. data lks (Steganography)

kita mendapatkan file datalks.zip lalu coba kita ekstrak



lalu membuat program ocr dengan python untuk membaca karakter dalam setiap gambar dimulai dari gambar 0 sampai dengan 130

```
GNU nano 7.2
import requests
import json
import sys

def ocr_space_file(filename, overlay=False, api_key='helloworld', language='eng', OCREngine=2):
    """ OCR.space API request with local file.
    Python3.5 - not tested on 2.7
    :param filename: Your file path & name.
    :param overlay: Is OCR.space overlay required in your response.
        Defaults to False.
    :param api_key: OCR.space API key.
        Defaults to 'helloworld'.
    :param language: Language code to be used in OCR.
        List of available language codes can be found on https://ocr.space/OCRAPI
        Defaults to 'en'.
    :return: Result in JSON format.
    """
    payload = {'isOverlayRequired': overlay,
               'apikey': api_key,
               'language': language,
               'OCREngine': OCREngine}
    with open(filename, 'rb') as f:
        r = requests.post('https://api.ocr.space/parse/image',
                          files={'filename': f},
                          data=payload,
                          )
    return r.content.decode()

def ocr_space_url(url, overlay=False, api_key='helloworld', language='eng'):
    """ OCR.space API request with remote file.
    Python3.5 - not tested on 2.7
    :param url: Image url.
    :param overlay: Is OCR.space overlay required in your response.
        Defaults to False.
    :param api_key: OCR.space API key.
        Defaults to 'helloworld'.
    :param language: Language code to be used in OCR.
        List of available language codes can be found on https://ocr.space/OCRAPI
        Defaults to 'en'.
    :return: Result in JSON format.
    """
    payload = {'url': url,
               'isOverlayRequired': overlay,
               'apikey': api_key,
               'language': language,
               }
    r = requests.post('https://api.ocr.space/parse/image',
                      data=payload,
                      )
    return r.content.decode()

# Use Examples!
# ocr_file = ocr_space_file(filename='sample_image.png', language='pol')
# test_url = ocr_space_url(url='http://i.imgur.com/13d15y.jpg')

strings=""
for i in range(131):
    print('gambar ke '+str(i))
    FilePath = "/home/ilalang/Downloads/data/img"+str(i)+".png"
    Engine = 2
    ocr_res = ''
    while len(ocr_res) < 2:
        ocr_res = json.loads(ocr_space_file(FilePath, api_key='K82695A3A588957', OCREngine=Engine))['ParsedResults'][0]['ParsedText']
        strings += ocr_res
    f = open("string_ocr_dari_depan.txt", "w")
    f.write(strings)
    f.close()
```

```
GNU nano 7.2
OCR.PY
with open(filename, 'rb') as f:
    r = requests.post('https://api.ocr.space/parse/image',
                      files={'filename': f},
                      data=payload,
                      )
    return r.content.decode()

def ocr_space_url(url, overlay=False, api_key='helloworld', language='eng'):
    """ OCR.space API request with remote file.
    Python3.5 - not tested on 2.7
    :param url: Image url.
    :param overlay: Is OCR.space overlay required in your response.
        Defaults to False.
    :param api_key: OCR.space API key.
        Defaults to 'helloworld'.
    :param language: Language code to be used in OCR.
        List of available language codes can be found on https://ocr.space/OCRAPI
        Defaults to 'en'.
    :return: Result in JSON format.
    """
    payload = {'url': url,
               'isOverlayRequired': overlay,
               'apikey': api_key,
               'language': language,
               }
    r = requests.post('https://api.ocr.space/parse/image',
                      data=payload,
                      )
    return r.content.decode()

# Use Examples!
# ocr_file = ocr_space_file(filename='sample_image.png', language='pol')
# test_url = ocr_space_url(url='http://i.imgur.com/13d15y.jpg')

strings=""
for i in range(131):
    print('gambar ke '+str(i))
    FilePath = "/home/ilalang/Downloads/data/img"+str(i)+".png"
    Engine = 2
    ocr_res = ''
    while len(ocr_res) < 2:
        ocr_res = json.loads(ocr_space_file(FilePath, api_key='K82695A3A588957', OCREngine=Engine))['ParsedResults'][0]['ParsedText']
        strings += ocr_res
    f = open("string_ocr_dari_depan.txt", "w")
    f.write(strings)
    f.close()
```

hasil dari program ocr :

*fJRMdPvURyRe9BC4n5ZpRgjzri8SzW7vjPXuxgZ1wNPoPXC7RJnxRHAtQ1t7N7vAsahEGRk7DPDRfsYwa
zZfgPpgF4FCD9LpPHEHESfNvqff2Czjp6ttMzpgF1PJ8vm2u72gjZbHXdNzupy4geCft4XtYPcjGHQLvpzVi6
rVqfUrp4S9XXJUETiA1agcdZTE4R1L4p7mSHAqBatZAMtsyKsTx33juFrjQtinMvzb4DfKt4HXgEGU4KPT6A
7iaqfXWwV8ct*

setelah kita analyze cipher diatas menggunakan base58 lalu coba kita decode dan muncul flag nya

LKS2023Malang{whY_whYYyY_d783fm}

```
(root@msi)-[/home/ilalang/Downloads]
# echo fJRmDPvUryRe9BCAn5ZpRgjzri8SzWVvjPXuxgZ1wNPopXC7RJnxRHATq1t7N7vAsahEGRk7DPDRfsYwazZfgPpgF4FCD9LpPHEHESfNvqff2Czjp6ttMzpgF1PJ3vm2u72gzjZbHXdZuppy4gcFct4tYpCj6HQLvpzV6rVqfUrp4S9XXJUEtiA1agcdZTEAR1L4p7mSHAqBatZAMtsyKsTx3j3uFrJqTinMvzb4DfKt4HXgEGU4KPT67iaqfXWw8Cct | base58 -d
I know it's hard to keep an open heart
When even friends seem out to harm you
But if you could heal a broken heart
Wouldn't time be out to charm you?

For you: LKS2023Malang[whY_whYYyY_d783fm]
(root@msi)-[/home/ilalang/Downloads]
```

7. Ferguso

mendapatkan satu gambar dengan ekstensi png lalu saya coba buka dengan zsteg

```

1  # -w -m -x -t -f /home/italang/Downloads
2  # -d zsteg -a ferguso-2.png
3  # -r rgb,lab,xy
4  # -r rgb,lab,xy
5  # -r rgb,lab,xy
6  # -r g,mb,xy
7  # -r b,mb,xy
8  # -r abgr,mb,xy
9  # -r b,lab,xy
10 # -r abgr,mb,xy
11 # -r f,lab,xy
12 # -r f,mb,xy
13 # -r g,lab,xy
14 # -r g,mb,xy
15 # -r b,lab,xy
16 # -r b,mb,xy
17 # -r abgr,lab,xy
18 # -r abgr,mb,xy
19 # -r f,lab,xy
20 # -r f,mb,xy
21 # -r g,lab,xy
22 # -r g,mb,xy
23 # -r b,lab,xy
24 # -r b,mb,xy
25 # -r abgr,lab,xy
26 # -r abgr,mb,xy
27 # -r f,lab,xy
28 # -r f,mb,xy
29 # -r g,lab,xy
30 # -r g,mb,xy
31 # -r b,lab,xy
32 # -r b,mb,xy
33 # -r abgr,lab,xy
34 # -r abgr,mb,xy
35 # -r f,lab,xy
36 # -r f,mb,xy
37 # -r g,lab,xy
38 # -r g,mb,xy
39 # -r b,lab,xy
40 # -r b,mb,xy
41 # -r abgr,lab,xy
42 # -r abgr,mb,xy
43 # -r f,lab,xy
44 # -r f,mb,xy
45 # -r g,lab,xy
46 # -r g,mb,xy
47 # -r b,lab,xy
48 # -r b,mb,xy
49 # -r abgr,lab,xy
50 # -r abgr,mb,xy
51 # -r f,lab,xy
52 # -r f,mb,xy
53 # -r g,lab,xy
54 # -r g,mb,xy
55 # -r b,lab,xy
56 # -r b,mb,xy
57 # -r abgr,lab,xy
58 # -r abgr,mb,xy
59 # -r f,lab,xy
60 # -r f,mb,xy
61 # -r g,lab,xy
62 # -r g,mb,xy
63 # -r b,lab,xy
64 # -r b,mb,xy
65 # -r abgr,lab,xy
66 # -r abgr,mb,xy
67 # -r f,lab,xy
68 # -r f,mb,xy
69 # -r g,lab,xy
70 # -r g,mb,xy
71 # -r b,lab,xy
72 # -r b,mb,xy
73 # -r abgr,lab,xy
74 # -r abgr,mb,xy
75 # -r f,lab,xy
76 # -r f,mb,xy
77 # -r g,lab,xy
78 # -r g,mb,xy
79 # -r b,lab,xy
80 # -r b,mb,xy
81 # -r abgr,lab,xy
82 # -r abgr,mb,xy
83 # -r f,lab,xy
84 # -r f,mb,xy
85 # -r g,lab,xy
86 # -r g,mb,xy
87 # -r b,lab,xy
88 # -r b,mb,xy
89 # -r abgr,lab,xy
90 # -r abgr,mb,xy
91 # -r f,lab,xy
92 # -r f,mb,xy
93 # -r g,lab,xy
94 # -r g,mb,xy
95 # -r b,lab,xy
96 # -r b,mb,xy
97 # -r abgr,lab,xy
98 # -r abgr,mb,xy
99 # -r f,lab,xy
100 # -r f,mb,xy
101 # -r g,lab,xy
102 # -r g,mb,xy
103 # -r b,lab,xy
104 # -r b,mb,xy
105 # -r abgr,lab,xy
106 # -r abgr,mb,xy
107 # -r f,lab,xy
108 # -r f,mb,xy
109 # -r g,lab,xy
110 # -r g,mb,xy
111 # -r b,lab,xy
112 # -r b,mb,xy
113 # -r abgr,lab,xy
114 # -r abgr,mb,xy
115 # -r f,lab,xy
116 # -r f,mb,xy
117 # -r g,lab,xy
118 # -r g,mb,xy
119 # -r b,lab,xy
120 # -r b,mb,xy
121 # -r abgr,lab,xy
122 # -r abgr,mb,xy
123 # -r f,lab,xy
124 # -r f,mb,xy
125 # -r g,lab,xy
126 # -r g,mb,xy
127 # -r b,lab,xy
128 # -r b,mb,xy
129 # -r abgr,lab,xy
130 # -r abgr,mb,xy
131 # -r f,lab,xy
132 # -r f,mb,xy
133 # -r g,lab,xy
134 # -r g,mb,xy
135 # -r b,lab,xy
136 # -r b,mb,xy
137 # -r abgr,lab,xy
138 # -r abgr,mb,xy
139 # -r f,lab,xy
140 # -r f,mb,xy
141 # -r g,lab,xy
142 # -r g,mb,xy
143 # -r b,lab,xy
144 # -r b,mb,xy
145 # -r abgr,lab,xy
146 # -r abgr,mb,xy
147 # -r f,lab,xy
148 # -r f,mb,xy
149 # -r g,lab,xy
150 # -r g,mb,xy
151 # -r b,lab,xy
152 # -r b,mb,xy
153 # -r abgr,lab,xy
154 # -r abgr,mb,xy
155 # -r f,lab,xy
156 # -r f,mb,xy
157 # -r g,lab,xy
158 # -r g,mb,xy
159 # -r b,lab,xy
160 # -r b,mb,xy
161 # -r abgr,lab,xy
162 # -r abgr,mb,xy
163 # -r f,lab,xy
164 # -r f,mb,xy
165 # -r g,lab,xy
166 # -r g,mb,xy
167 # -r b,lab,xy
168 # -r b,mb,xy
169 # -r abgr,lab,xy
170 # -r abgr,mb,xy
171 # -r f,lab,xy
172 # -r f,mb,xy
173 # -r g,lab,xy
174 # -r g,mb,xy
175 # -r b,lab,xy
176 # -r b,mb,xy
177 # -r abgr,lab,xy
178 # -r abgr,mb,xy
179 # -r f,lab,xy
180 # -r f,mb,xy
181 # -r g,lab,xy
182 # -r g,mb,xy
183 # -r b,lab,xy
184 # -r b,mb,xy
185 # -r abgr,lab,xy
186 # -r abgr,mb,xy
187 # -r f,lab,xy
188 # -r f,mb,xy
189 # -r g,lab,xy
190 # -r g,mb,xy
191 # -r b,lab,xy
192 # -r b,mb,xy
193 # -r abgr,lab,xy
194 # -r abgr,mb,xy
195 # -r f,lab,xy
196 # -r f,mb,xy
197 # -r g,lab,xy
198 # -r g,mb,xy
199 # -r b,lab,xy
200 # -r b,mb,xy
201 # -r abgr,lab,xy
202 # -r abgr,mb,xy
203 # -r f,lab,xy
204 # -r f,mb,xy
205 # -r g,lab,xy
206 # -r g,mb,xy
207 # -r b,lab,xy
208 # -r b,mb,xy
209 # -r abgr,lab,xy
210 # -r abgr,mb,xy
211 # -r f,lab,xy
212 # -r f,mb,xy
213 # -r g,lab,xy
214 # -r g,mb,xy
215 # -r b,lab,xy
216 # -r b,mb,xy
217 # -r abgr,lab,xy
218 # -r abgr,mb,xy
219 # -r f,lab,xy
220 # -r f,mb,xy
221 # -r g,lab,xy
222 # -r g,mb,xy
223 # -r b,lab,xy
224 # -r b,mb,xy
225 # -r abgr,lab,xy
226 # -r abgr,mb,xy
227 # -r f,lab,xy
228 # -r f,mb,xy
229 # -r g,lab,xy
230 # -r g,mb,xy
231 # -r b,lab,xy
232 # -r b,mb,xy
233 # -r abgr,lab,xy
234 # -r abgr,mb,xy
235 # -r f,lab,xy
236 # -r f,mb,xy
237 # -r g,lab,xy
238 # -r g,mb,xy
239 # -r b,lab,xy
240 # -r b,mb,xy
241 # -r abgr,lab,xy
242 # -r abgr,mb,xy
243 # -r f,lab,xy
244 # -r f,mb,xy
245 # -r g,lab,xy
246 # -r g,mb,xy
247 # -r b,lab,xy
248 # -r b,mb,xy
249 # -r abgr,lab,xy
250 # -r abgr,mb,xy
251 # -r f,lab,xy
252 # -r f,mb,xy
253 # -r g,lab,xy
254 # -r g,mb,xy
255 # -r b,lab,xy
256 # -r b,mb,xy
257 # -r abgr,lab,xy
258 # -r abgr,mb,xy
259 # -r f,lab,xy
260 # -r f,mb,xy
261 # -r g,lab,xy
262 # -r g,mb,xy
263 # -r b,lab,xy
264 # -r b,mb,xy
265 # -r abgr,lab,xy
266 # -r abgr,mb,xy
267 # -r f,lab,xy
268 # -r f,mb,xy
269 # -r g,lab,xy
270 # -r g,mb,xy
271 # -r b,lab,xy
272 # -r b,mb,xy
273 # -r abgr,lab,xy
274 # -r abgr,mb,xy
275 # -r f,lab,xy
276 # -r f,mb,xy
277 # -r g,lab,xy
278 # -r g,mb,xy
279 # -r b,lab,xy
280 # -r b,mb,xy
281 # -r abgr,lab,xy
282 # -r abgr,mb,xy
283 # -r f,lab,xy
284 # -r f,mb,xy
285 # -r g,lab,xy
286 # -r g,mb,xy
287 # -r b,lab,xy
288 # -r b,mb,xy
289 # -r abgr,lab,xy
290 # -r abgr,mb,xy
291 # -r f,lab,xy
292 # -r f,mb,xy
293 # -r g,lab,xy
294 # -r g,mb,xy
295 # -r b,lab,xy
296 # -r b,mb,xy
297 # -r abgr,lab,xy
298 # -r abgr,mb,xy
299 # -r f,lab,xy
300 # -r f,mb,xy
301 # -r g,lab,xy
302 # -r g,mb,xy
303 # -r b,lab,xy
304 # -r b,mb,xy
305 # -r abgr,lab,xy
306 # -r abgr,mb,xy
307 # -r f,lab,xy
308 # -r f,mb,xy
309 # -r g,lab,xy
310 # -r g,mb,xy
311 # -r b,lab,xy
312 # -r b,mb,xy
313 # -r abgr,lab,xy
314 # -r abgr,mb,xy
315 # -r f,lab,xy
316 # -r f,mb,xy
317 # -r g,lab,xy
318 # -r g,mb,xy
319 # -r b,lab,xy
320 # -r b,mb,xy
321 # -r abgr,lab,xy
322 # -r abgr,mb,xy
323 # -r f,lab,xy
324 # -r f,mb,xy
325 # -r g,lab,xy
326 # -r g,mb,xy
327 # -r b,lab,xy
328 # -r b,mb,xy
329 # -r abgr,lab,xy
330 # -r abgr,mb,xy
331 # -r f,lab,xy
332 # -r f,mb,xy
333 # -r g,lab,xy
334 # -r g,mb,xy
335 # -r b,lab,xy
336 # -r b,mb,xy
337 # -r abgr,lab,xy
338 # -r abgr,mb,xy
339 # -r f,lab,xy
340 # -r f,mb,xy
341 # -r g,lab,xy

```

saya coba decode text diatas menggunakan base64

"ewogIGtleTphZXNjYmMzNTYtTWFsYW5nS290YVBlbGFqYXJCYW55YWtLYW1wdXMxMjMKICBjaXBoZ
XI6MjIwNjIzNUY1QTc0NDQ4MzI3O
DAxRTdGmkE0Q0QwRkUyNjBFNkMwNTUxM0E2NDQ4RUQ4RDY3RDU2MkYwNDZBRgp9"


```
(root@msi) - [ /home/ilalang/Downloads ]
# echo ewogIGtleTphZXNjYmMzNTYtTWFsYW5nS290YVBlbGFqYXJCYW55YWtLYW1wdXMxMjMKICBjaXB0ZXI6MjIwNjIzNUY1QTc0NDQ4MzI3ODAxRTdGMkE0Q0QwRkUyNjBFNkMwNTUxM0E2NDQ4RUQ4RDY3RDU2MkYwNDZBRgp9 | base64 -d
{
  key:aescbc356-MalangKotaPelajarBanyakKampus123
  cipher:2206235F5A74448327801E7F2A4CD0FE260E6C05513A6448ED8D67D562F046AF
}
```

decode lagi cipher diatas menggunakan aescbc 256 dengan key
MalangKotaPelajaraBanyakKampus123

AES Online Decryption

Enter text to be Decrypted

2206235F5A74448327801E7F2A4CD0FE260E6C05513A6448ED8D67D562F046AF

Input Text Format: ☐ Base64 ☒ Hex

Select Cipher Mode of Decryption

CBC

Enter IV Used During Encryption(Optional)

Enter initialization vector

Key Size in Bits

256

Enter Secret Key used for Encryption

MalangKotaPelajarBanyakKampus123

Decrypt

AES Decrypted Output (Base64):

TtTMjAyM0lhbGFuZ3toMWQzXzB2M3JfaW00ZzN9

Decode to Plain Text

LKS2023Malang{h1d3_0v3r_im4g3}

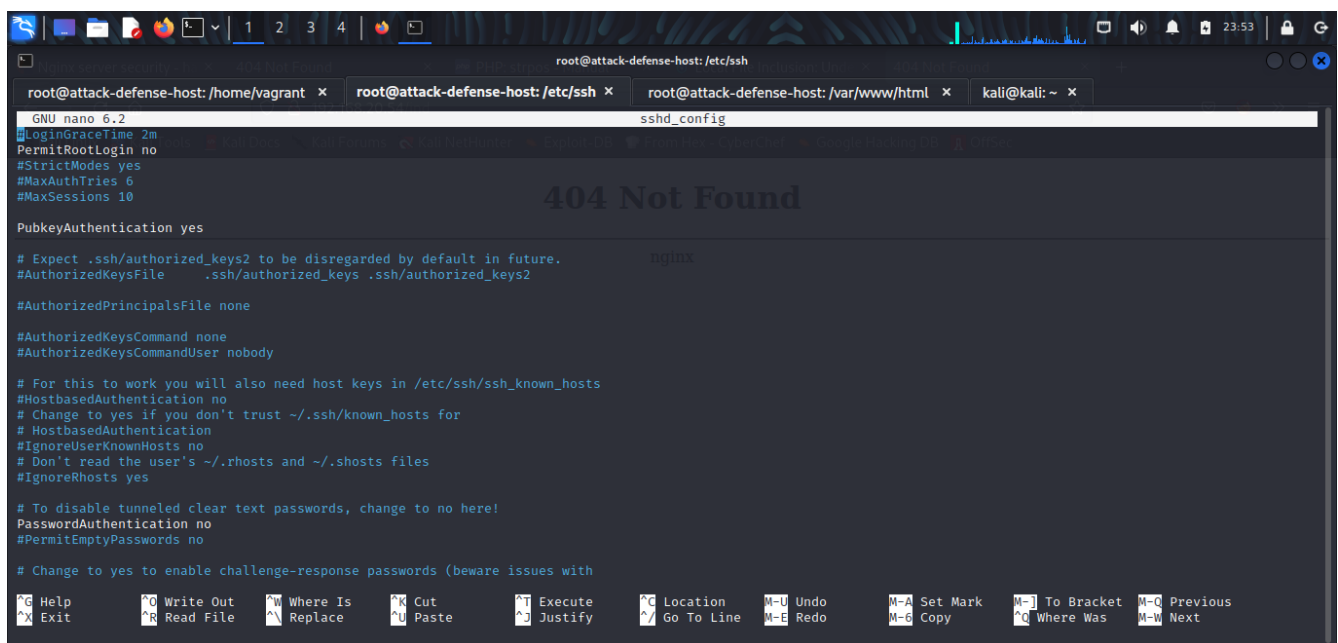
BAB B: Defense

1. Password

- mengganti password default untuk ubuntu dan vargant

2. SSH

- disable PermitRootLogin
untuk melarang login lewat ssh menggunakan root
- Disable passwordAuthentication
Agar login hanya bisa menggunakan public key
- Enable PublicKeyAuthentication
untuk enable public key



```
GNU nano 6.2 sshd_config
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

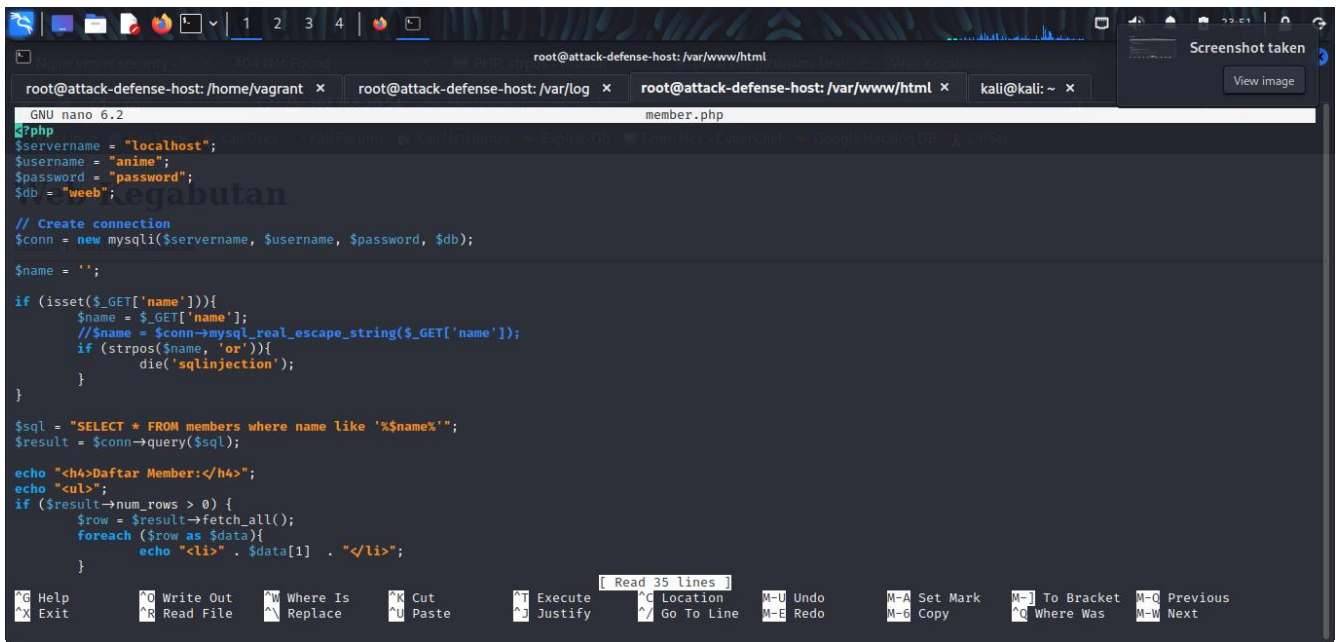
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify
^C Location  ^_/ Go To Line ^M-U Undo     ^M-A Set Mark ^M-J To Bracket ^M-Q Previous
^M-G Where Was ^M-W Next
```

3. Path WEB

ada celah web yaitu LFI dan sqlinjection

jadi kami menutup celah tersebut menggunakan function strpos() untuk menemukan karakter yang dilarang, lalu jika karakter yang dilarang terdeteksi, maka php akan menjalankan perintah die() atau menghentikan semua perintah



```
GNU nano 6.2 member.php
<?php
$servername = "localhost";
$username = "anime";
$password = "password";
$db = "weeb";

// Create connection
$conn = new mysqli($servername, $username, $password, $db);

$name = '';

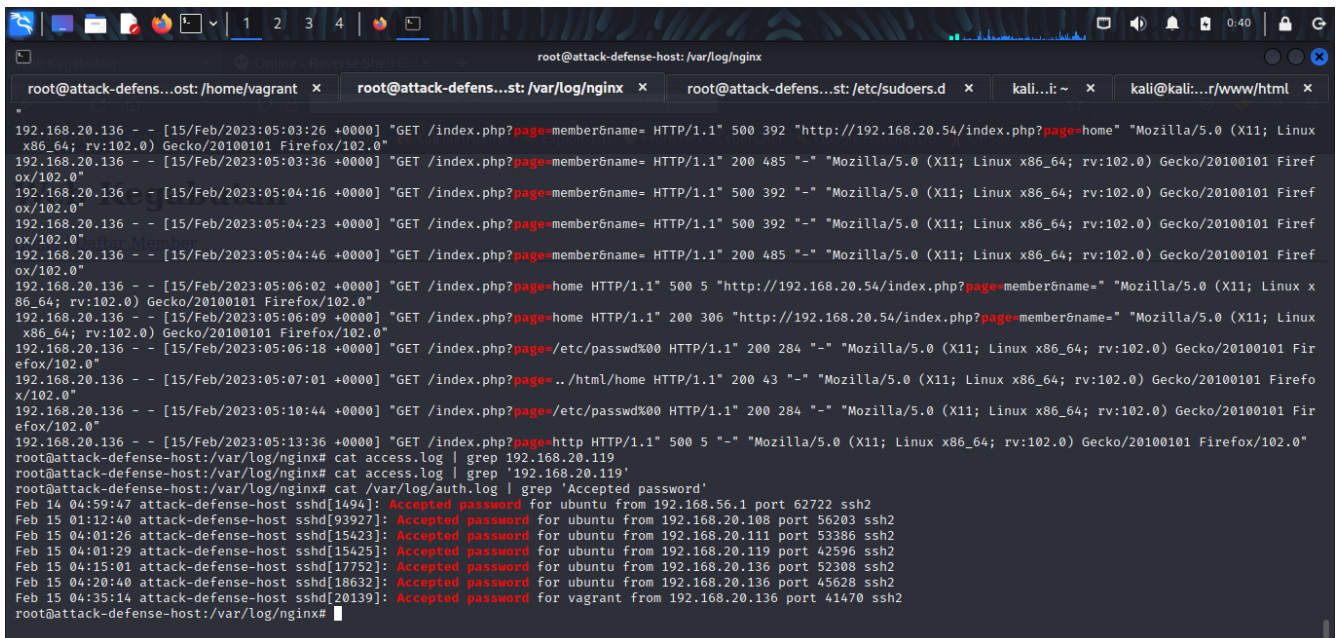
if (isset($_GET['name'])) {
    $name = $_GET['name'];
    // $name = $conn->mysql_real_escape_string($_GET['name']);
    if (strpos($name, 'or')) {
        die('sqlinjection');
    }
}

$sql = "SELECT * FROM members where name like '%$name%'";
$result = $conn->query($sql);

echo "<h4>Daftar Member:</h4>";
echo "<ul>";
if ($result->num_rows > 0) {
    $row = $result->fetch_all();
    foreach ($row as $data) {
        echo "<li>" . $data[1] . "</li>";
    }
}
```

5. Monitoring

kami menggunakan command tail -f /var/log/auth.log



```
192.168.20.136 - - [15/Feb/2023:05:03:26 +0000] "GET /index.php?page=member&name= HTTP/1.1" 500 392 "http://192.168.20.54/index.php?page=home" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:03:36 +0000] "GET /index.php?page=member&name= HTTP/1.1" 200 485 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:04:16 +0000] "GET /index.php?page=member&name= HTTP/1.1" 500 392 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:04:23 +0000] "GET /index.php?page=member&name= HTTP/1.1" 500 392 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:04:46 +0000] "GET /index.php?page=member&name= HTTP/1.1" 200 485 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:06:02 +0000] "GET /index.php?page=home HTTP/1.1" 500 5 "http://192.168.20.54/index.php?page=member&name=" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:06:09 +0000] "GET /index.php?page=home HTTP/1.1" 200 306 "http://192.168.20.54/index.php?page=member&name=" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:06:18 +0000] "GET /index.php?page=/etc/passwd%00 HTTP/1.1" 200 284 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:07:01 +0000] "GET /index.php?page=/html/home HTTP/1.1" 200 43 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:10:44 +0000] "GET /index.php?page=/etc/passwd%00 HTTP/1.1" 200 284 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.20.136 - - [15/Feb/2023:05:13:36 +0000] "GET /index.php?page=http HTTP/1.1" 500 5 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
root@attack-defense-host:/var/log/nginx# cat access.log | grep 192.168.20.119
root@attack-defense-host:/var/log/nginx# cat access.log | grep '192.168.20.119'
root@attack-defense-host:/var/log/nginx# cat /var/log/auth.log | grep 'Accepted password'
Feb 15 04:59:47 attack-defense-host sshd[1494]: Accepted password for ubuntu from 192.168.56.1 port 62722 ssh2
Feb 15 01:12:40 attack-defense-host sshd[93927]: Accepted password for ubuntu from 192.168.20.108 port 56203 ssh2
Feb 15 04:01:26 attack-defense-host sshd[15423]: Accepted password for ubuntu from 192.168.20.111 port 53386 ssh2
Feb 15 04:01:29 attack-defense-host sshd[15425]: Accepted password for ubuntu from 192.168.20.119 port 42596 ssh2
Feb 15 04:15:01 attack-defense-host sshd[17752]: Accepted password for ubuntu from 192.168.20.136 port 52308 ssh2
Feb 15 04:20:40 attack-defense-host sshd[18632]: Accepted password for ubuntu from 192.168.20.136 port 45628 ssh2
Feb 15 04:35:14 attack-defense-host sshd[20139]: Accepted password for vagrant from 192.168.20.136 port 41470 ssh2
root@attack-defense-host:/var/log/nginx#
```

BAB C: ATTACK

- Pertama kita cek celah di port http ternyata ada celah untuk lfi tapi tidak bisa digunakan tapi bisa menggunakan celah sql injection dengan memasukkan payload di parameter get, nama parameter nya yaitu 'id'
- Setelah itu kita menemukan vulnerability yaitu weak password di semua server yaitu password default vagrant
- Privilege escalation langsung menggunakan sudo karna vagrant merupakan sudoers no password

The screenshot displays a web application interface for a CTF challenge. The top navigation bar is red and contains links: Users, Teams, Scoreboard, Challenges, Notifications, Team, Profile, and Settings. The main heading is "Challenges". Below this, the "Attack Defence" category is selected, showing a grid of 12 server challenge cards. Each card is green and contains the server name, IP address, and a score. The servers are arranged in a 3x4 grid. At the bottom, there is a file explorer showing two files: "member.php" and "index.php". The interface is powered by CTFd.

Server	IP Address	Score
Server 1	192.168.20.51	999
Server 1 Root	192.168.20.51	999
Server 2	192.168.20.52	1000
Server 3	192.168.20.53	1000
Server 4	192.168.20.54	1000
Server 5	192.168.20.55	1000
Server 6	192.168.20.56	1000
Server 2 Root	192.168.20.52	1000
Server 3 Root	192.168.20.53	1000
Server 4 Root	192.168.20.54	1000
Server 5 Root	192.168.20.55	1000
Server 6 Root	192.168.20.56	1000