# BÀI TẬP CTF

## Bảo mật web và ứng dụng – NT213.M21.ANTN

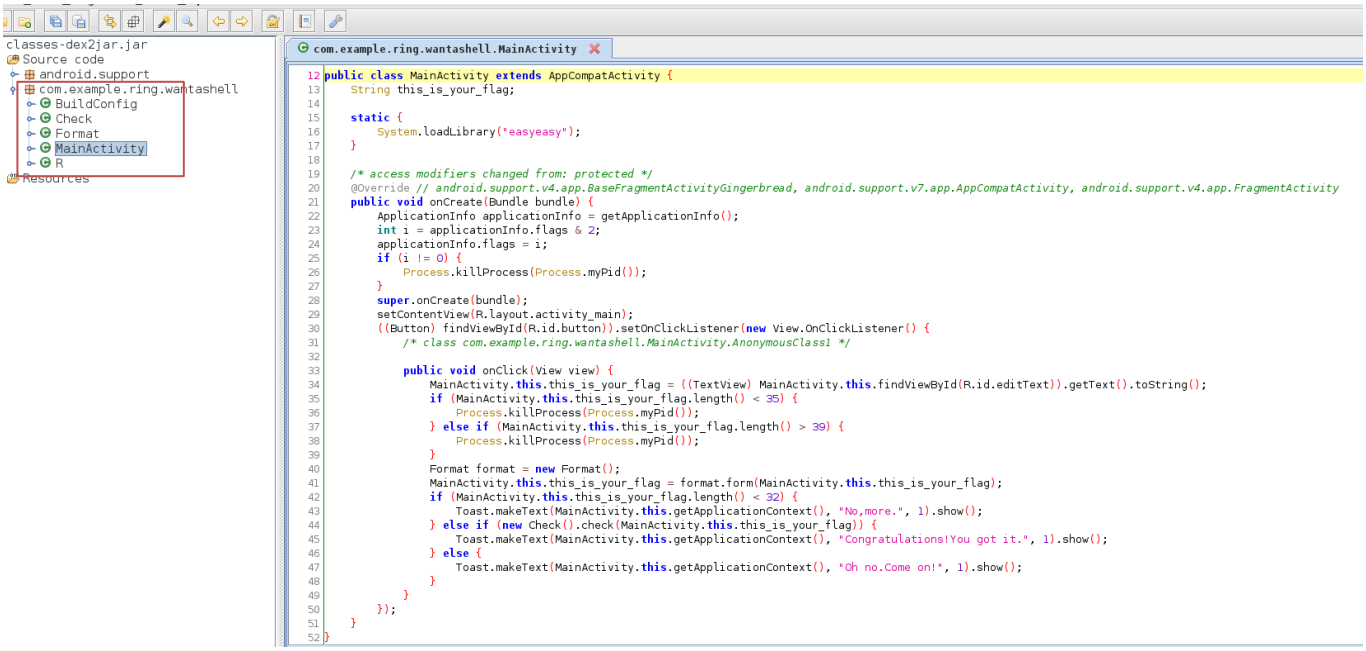
**Giảng viên hướng dẫn:** *Đỗ Hoàng Hiển*

**Sinh viên thực hiện:** *19520199 – Lê Tôn Nhân*

### Challenge 5: WantAShell

- Cài đặt ứng dụng, ở đây em sử dụng LDPlayer để chạy ứng dụng
- Ta thấy ứng dụng đơn giản gồm một input nhập vào. Không hề có nút submit hay gì, khi ta nhập vào bất kỳ gì đó và nhấn enter thì chương trình sẽ thoát


- Giải nén file WantAShell.apk, sau đó sử dụng dex2jax để chuyển file class.dex thành file jar
- Sử dụng jadx-gui để dịch ngược ứng dụng và xem các logic cơ bản của nó




- Có 5 mô đun java. Ta bắt đầu kiểm tra với MainActivity.
- Ở MainActivity ta thấy có hàm onclick trong hàm oncreat, do đó có thể người tạo ra challenge đã quên thêm button vào)

```
public void onCreate(Bundle bundle) {
    ApplicationInfo applicationInfo = getApplicationInfo();
    int i = applicationInfo.flags & 2;
    applicationInfo.flags = i;
    if (i != 0) {
        Process.killProcess(Process.myPid());
    }
    super.onCreate(bundle);
    setContentView(R.layout.activity_main);
    ((Button) findViewById(R.id.button)).setOnClickListener(new View.OnClickListener() {
        /* class com.example.ring.wantashell.MainActivity.AnonymousClass1 */

        public void onClick(View view) {
            MainActivity.this.this_is_your_flag = ((TextView) MainActivity.this.findViewById(R.id.editText)).getText().toString();
            if (MainActivity.this.this_is_your_flag.length() < 35) {
                Process.killProcess(Process.myPid());
            } else if (MainActivity.this.this_is_your_flag.length() > 39) {
                Process.killProcess(Process.myPid());
            }
            Format format = new Format();
            MainActivity.this.this_is_your_flag = format.form(MainActivity.this.this_is_your_flag);
            if (MainActivity.this.this_is_your_flag.length() < 32) {
                Toast.makeText(MainActivity.this.getApplicationContext(), "No,more.", 1).show();
            } else if (new Check().check(MainActivity.this.this_is_your_flag)) {
                Toast.makeText(MainActivity.this.getApplicationContext(), "Congratulations!You got it.", 1).show();
            } else {
                Toast.makeText(MainActivity.this.getApplicationContext(), "Oh no.Come on!", 1).show();
            }
        }
    });
```

- Ở hàm onclick đầu tiên ta thấy nó kiểm tra chuỗi ta nhập vào. Nếu chiều dài nhỏ hơn 35 hoặc lớn hơn 39 thì sẽ dừng tiến trình

```
public void onCreate(Bundle bundle) {
    ApplicationInfo applicationInfo = getApplicationInfo();
    int i = applicationInfo.flags & 2;
    applicationInfo.flags = i;
    if (i != 0) {
        Process.killProcess(Process.myPid());
    }
    super.onCreate(bundle);
    setContentView(R.layout.activity_main);
    ((Button) findViewById(R.id.button)).setOnClickListener(new View.OnClickListener() {
        /* class com.example.ring.wantashell.MainActivity.AnonymousClass1 */

        public void onClick(View view) {
            MainActivity.this.this_is_your_flag = ((TextView) MainActivity.this.findViewById(R.id.editText)).getText().toString();
            if (MainActivity.this.this_is_your_flag.length() < 35) {
                Process.killProcess(Process.myPid());
            } else if (MainActivity.this.this_is_your_flag.length() > 39) {
                Process.killProcess(Process.myPid());
            }
            Format format = new Format();
            MainActivity.this.this_is_your_flag = format.form(MainActivity.this.this_is_your_flag);
            if (MainActivity.this.this_is_your_flag.length() < 32) {
                Toast.makeText(MainActivity.this.getApplicationContext(), "No,more.", 1).show();
            } else if (new Check().check(MainActivity.this.this_is_your_flag)) {
                Toast.makeText(MainActivity.this.getApplicationContext(), "Congratulations!You got it.", 1).show();
            } else {
                Toast.makeText(MainActivity.this.getApplicationContext(), "Oh no.Come on!", 1).show();
            }
        }
    });
```

- Rồi sau đó gọi format.form để chỉ lấy substring(5,38) của biến this_is_your_flag mà ta nhập vào. Nếu nhỏ hơn 32 thì xuất ra thông báo **No more**.

```
public void onClick(View view) {
    MainActivity.this.this_is_your_flag = ((TextView) MainActivity.this.findViewById(R.id.editText)).getText().toString();
    if (MainActivity.this.this_is_your_flag.length() < 35) {
        Process.killProcess(Process.myPid());
    } else if (MainActivity.this.this_is_your_flag.length() > 39) {
        Process.killProcess(Process.myPid());
    }
    Format format = new Format();
    MainActivity.this.this_is_your_flag = format.form(MainActivity.this.this_is_your_flag);
    if (MainActivity.this.this_is_your_flag.length() < 32) {
        Toast.makeText(MainActivity.this.getApplicationContext(), "No,more.", 1).show();
    } else if (new Check().check(MainActivity.this.this_is_your_flag)) {
        Toast.makeText(MainActivity.this.getApplicationContext(), "Congratulations!You got it.", 1).show();
    } else {
        Toast.makeText(MainActivity.this.getApplicationContext(), "Oh no.Come on!", 1).show();
    }
```

com.example.ring.wantashell.MainActivity | com.example.ring.wantash

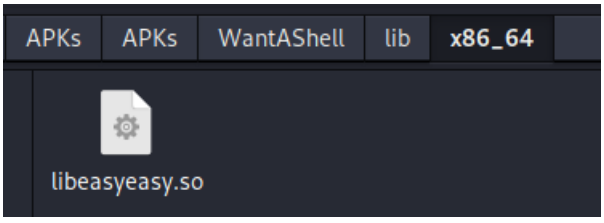
```
1  package com.example.ring.wantashell;
2
3  public class Format {
4      /* access modifiers changed from: protected */
5      public String f0rm(String str) {
6          return str.substring(5, 37);
7      }
8
9      /* access modifiers changed from: protected */
10     public String f0lm(String str) {
11         return str.substring(5, 36);
12     }
13
14     /* access modifiers changed from: protected */
15     public String form(String str) {
16         return str.substring(5, 38);
17     }
18
19     /* access modifiers changed from: protected */
20     public String forn(String str) {
21         return str.substring(5, 39);
22     }
23 }
```

- Kết hợp các điều kiên ở trên lại ta suy ra được chuỗi của ta chỉ có thể có độ dài là 38.
- Để xuất ra chuỗi "**Congratulations! You got it**" thì hàm check sẽ được gọi và kiểm tra. Trong hàm này sẽ kiểm tra nhiều lần và trong các hàm con thì mỗi hàm kiểm tra 1 lần và trả về true

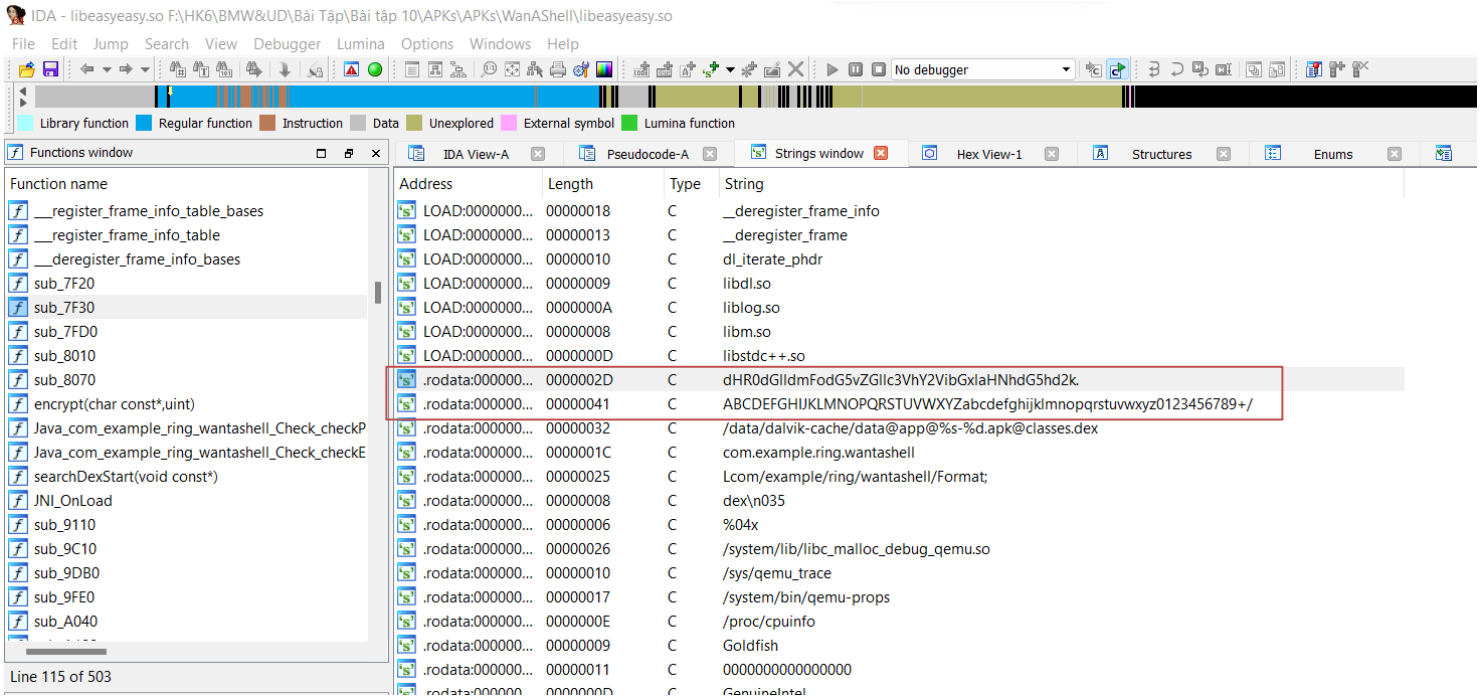
```
package com.example.ring.wantashell;

import java.io.File;

public class Check {
    private static String[] known_pipes = {"/dev/socket/qemud", "/dev/qemu_pipe"};
    String emulator = checkPipes();

    private native boolean checkPasswd(String str);

    public static String checkPipes() {
        for (int i = 0; i < known_pipes.length; i++) {
            if (new File(known_pipes[i]).exists()) {
                return "true";
            }
        }
        return "false";
    }

    /* access modifiers changed from: package-private */
    public boolean check(String str) {
        if (checkEmulator(this.emulator)) {
            return false;
        }
        return checkPasswd(str);
    }

    /* access modifiers changed from: protected */
    public native boolean checkEmulator(String str);
}
```

- Sử dụng IDA để phân tích thư viện được sử dụng. Ta sẽ thấy file thư viện trong lib -> x86_64

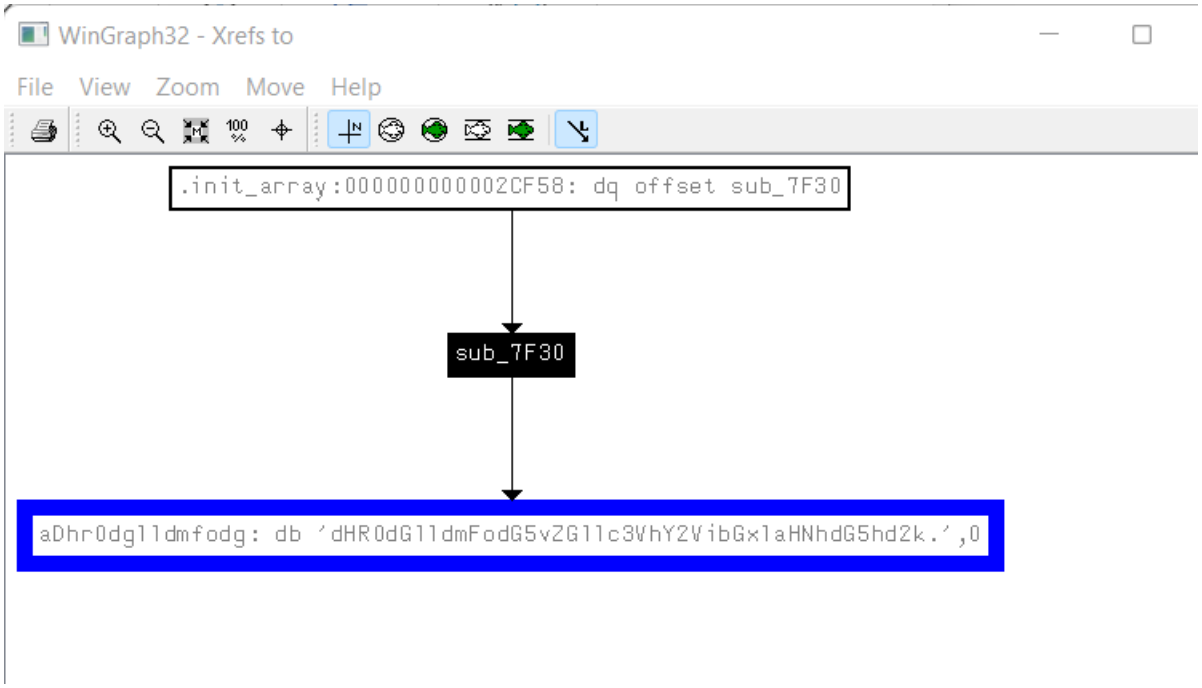
```
static {
    System.loadLibrary("easyeasy");
}
```




- Sử dụng **String windows** trong IDA ta thấy có các chuỗi khá đặc biệt



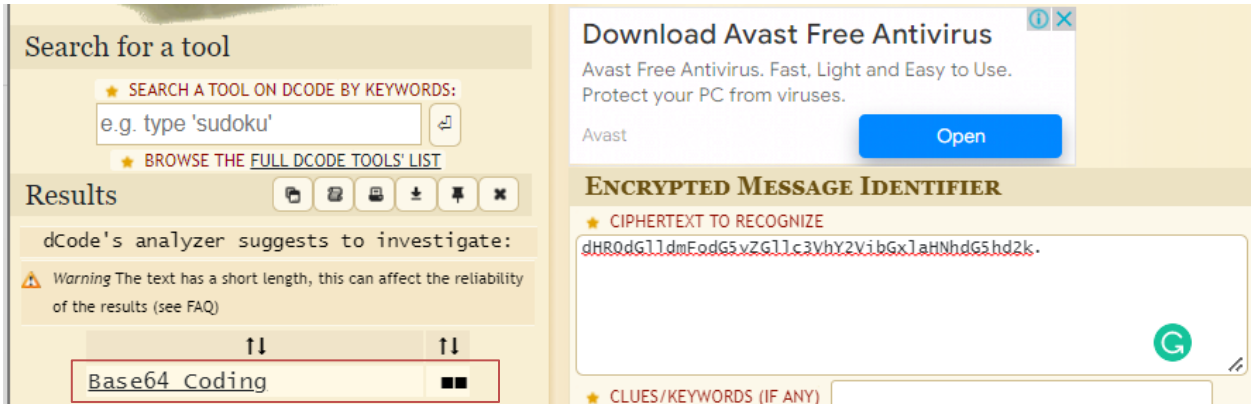

- Sử dụng **Xrefs graph to** để biết được các hàm liên quan đến chuỗi này




- Ta tìm được hàm sub_7F30. Tiến hành phân tích hàm đó bằng mã giả (sử dụng tab hoặc f5 để chuyển sang mã giả)

```
Data  Unexplored  External symbol  Lumina function
IDA View-A | Pseudocode-A | Strings window | Hex View-1 | Structures | Enums | Imports | Exports
 1 unsigned __int64 sub_7F30()
 2 {
 3   char v1[8]; // [rsp+8h] [rbp-30h] BYREF
 4   char v2[8]; // [rsp+10h] [rbp-28h] BYREF
 5   unsigned __int64 v3; // [rsp+18h] [rbp-20h]
 6
 7   v3 = __readfsqword(0x28u);
 8   sub_B400(&secret, "dHR0dGlldmFodG5vZGllc3VhY2VibGxlaHNhdG5hd2k.", v1);
 9   __cxa_atexit((void (__fastcall *)(void *))sub_B750, &secret, &off_2E000);
10   sub_B400(&qword_2E188, "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/", v2);
11   __cxa_atexit((void (__fastcall *)(void *))sub_B750, &qword_2E188, &off_2E000);
12   return __readfsqword(0x28u);
13 }
```

- Ta thấy chuỗi này có liên quan đến **secret** do đó ta có ý định sẽ thử **decode** chuỗi này
- Sử dụng tool **dcode** ( dcode.fr/cipher-identifier ) để detect loại mã hóa này. Ta nhận được kết quả cao nhất là base64




- Decode base64 đoạn mã này (thay dấu “.” ở cuối bằng dấu “=” vì tiêu chuẩn của base64 không có chứa dấu “.”) ta thu được chuỗi như bên dưới

```
┌──(kali㉿kali)-[~/…/Bai Tap 10/APKs/APKs/WantAShell]
└─$ echo "dHR0dG11dmFodG5vZG11c3VhY2VibGxlaHNhdG5hd2k=" | base64 -d
ttttmuvahtnodmusuacebllehsatnawi
```

- Đảo ngược chuỗi vừa decode sử dụng lệnh **rev** ta thu được flag thành công

```
┌──(kali㉿kali)-[~/…/Bai Tap 10/APKs/APKs/WantAShell]
└─$ echo "dHR0dG11dmFodG5vZG11c3VhY2VibGxlaHNhdG5hd2k=" | base64 -d | rev
iwantashellbecausumdonthavumtttt
```

**FLAG: iwantashellbecausumdonthavumtttt**