周报

本工具仅供参考使用

网络安全竞赛，破解题目获取"Flag"。

- **模式**：解题（逆向/Web/密码学）、攻防（实时攻防）。
- CTF主要分为五个方向，Web、pwn、crypto、misc和reverse（逆向）

CTF解题思路 - Misc

这类题目范围很宽泛，主要包括信息隐写、流量分析等等内容，也成为主办方脑洞的宣泄地。参赛者需要充分发挥"大脑风暴"，寻找对应题目的flag字符串，并提交。
主要知识点
隐写术
网络流量分析
网络编程与算法
社会工程学
数据恢复与取证分析

MISC题目通用方法：

经验+尝试+脑洞->flag

面对已知类型的题目，回顾做题思路

面对未知类型的题目，善于观察，寻找相关资料，尝试。

JSFuck的基本原理

JSFuck仅用 [ ] ( ) ! + 六个字符。
特征：全部类似这样的[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(![]+[])[!+[]+!+[]]][([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(![]+[])[!+[]+!+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(![]+[])[!+[]+!+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(![]+[])[!+[]+!+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(![]+[])[!+[]+!+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]((!![]+[])[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+([][[]]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+!+[]]+ 序列!

为什么需要 ()?
函数调用：JSFuck通常需要构造一个函数（如 eval 或 Function），然后显式调用它才能执行代码。

CTF实战中的意义
验证执行：题目通常要求代码实际执行（如弹窗、读取flag），缺少 () 会导致代码未触发。

常见陷阱：有时题目可能故意省略 () 作为考察点，需结合上下文判断。

在CTF（Capture The Flag）比赛中，图片高度隐写和字符替换是两类常见的隐写技术（Steganography），常用于在看似普通的数据中隐藏关键信息（如Flag）。以下是对这两种技术的详细介绍：

** 图片高度隐写**

原理

图片文件（如PNG、BMP等）的格式中通常包含一个文件头，其中存储了图片的元数据，如**宽度**和**高度**。通过**修改图片文件的高度值**，可以欺骗图片查看器仅显示部分内容，而实际文件中可能包含隐藏的数据（例如在不可见区域存储Flag）。

常见操作

1. **文件结构分析**
   - PNG文件：高度信息存储在IHDR数据块中（前8字节为宽度，后8字节为高度）。
   - BMP文件：高度信息在文件头的第22-25字节（小端存储）。
   - 使用工具（如`winhex、010 Editor`）直接修改高度值。
2. **隐藏数据提取**
   - 当高度值被修改为更小的数值时，图片显示不完整，但剩余数据可能包含隐藏信息。
   - 通过恢复正确的高度值或使用工具（如`pngcheck`、`exiftool`）分析文件结构，可以显示完整图片。
3. **CRC校验修复**
   - PNG文件的IHDR块包含CRC校验码。修改高度后，可能需要重新计算CRC值以修复图片（工具如`pngcsum`）。

实战案例

- **题目示例**
  给定一个被修改高度的PNG图片，图片显示不完整。通过十六进制编辑器修改高度值（或修复CRC），完整图片中可能包含Flag。

  工具推荐
  - `binwalk`（kali自带）：扫描文件中的隐藏数据。
  - `stegsolve`：逐像素分析图片的可疑区域。
  - 这是下载stegsolve的下载链接：https://pan.baidu.com/s/1UVS1LYVksQydxk3jeOqHkQ

    提取码：ip8v
  - **010 Editor**：解析文件结构（PE/ELF）、脚本自动化，适合CTF逆向分析。

    官网: https://www.sweetscape.com/010editor/
  - **WinHex**：磁盘编辑/数据恢复，操作简单，适合快速处理。

    官网: https://www.x-ways.net/winhex/

** 字符替换隐写**

原理

通过替换文本中的特定字符（如字母、符号、空格等）来隐藏信息。常见的替换方式包括：

1. **字符替换**
   把某一个字符替换成空格方便查看。
2. **编码替换**
   将字符替换为其他编码格式（如二进制、十六进制、Base64）。

如何应对CTF中的隐写题?

1. **文件头检查**
   使用 `file` 命令或 `hexdump` 检查文件是否被篡改。
2. **元数据分析**
   通过 `exiftool` 查看图片/文件的元数据。
3. **数据提取**
   使用 `dd` 或 `foremost` 从文件中分离隐藏数据。
4. **自动化工具**
   尝试 `stegoveritas`（自动检测图片隐写）或 `zsteg`（针对PNG/BMP的隐写工具）。

总结

- **图片高度隐写**：通过修改图片的高度值隐藏数据，需结合文件格式分析和校验修复。
- **字符替换隐写**：通过替换字符或插入不可见编码隐藏信息，依赖文本分析和编码转换。

在CTF比赛中，这两种技术常结合其他漏洞（如加密、逆向）出现，需要灵活运用多种工具和脚本进行深入分析。

HTML实体编码
简述：

字符实体是用一个编号写入HTML代码中来代替一个字符，在使用浏览器访问网页时会将这个编号解析还原为字符以供阅读。

原文链接：https://blog.csdn.net/m0_74077634/article/details/142101163

ascll码

ASCII（American Standard Code for Information Interchange，美国信息交换标准代码）是一种最基础的**字符编码标准**，诞生于1960年代。它的作用是将**字母、数字、符号**等字符映射为计算机能理解的**二进制数值**（0和1的组合）。

```
数字0~9对应的ASCII码（十进制）为“48”~“57”

大写字母A~Z对应的ASCII码（十进制）为“65”~“90”

小写字母a~z对应的百ASCII码（十进制）为"97"~“122”
```

案例1：数字转ASCII

1. **题目数据**：`67 84 70 123 65 83 67 73 73 95 82 48 99 107 115 125`
2. **解题步骤**：
   - 将每个十进制数转换为ASCII字符：

     ```
     67,84,70,123,65,83,67,73,73,95,82,48,99,107,115,125
     ```

   - **结果**：`CTF{ASCII_R0cks}`

https://blog.csdn.net/Henry_Mo_Fang/article/details/131522495?ops_request_misc=&request_id=&biz_id=102&utm_term=ascll%E8%A1%A8&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-131522495.nonecase&spm=1018.2226.3001.4187

Base64编码的特征**

1. **字符集固定**
   - 仅包含 **64个可打印字符**：`A-Z`、`a-z`、`0-9`、`+`、`/`。
   - 结尾可能带有 **等号（`=`）** 作为填充符（如`...==`或`...=`）。
2. **长度规则**
   - 编码后的字符串长度通常是 **4的倍数**（不足时用`=`填充）。
   - 例如：`"flag"` → `"ZmxhZw=="`（长度为8）

RkxBR3tCNDUzNjRfMTVfQ09PTH0=

凯撒密码的特征

1. **加密规则简单**
   每个字母在字母表中**固定偏移**（如右移3位，`A→D`，`B→E`，`Z→C`）。
2. **仅处理字母**
   非字母字符（如数字、符号）通常**保持原样。**
3. **可暴力破解**
   由于偏移量仅有25种可能（1-25），可遍历所有情况找Flag。
4. 常见变种
   - **ROT13**：偏移13位（字母表对称，加密=解密）。
   - **自定义偏移**：如偏移5、7位等。
   - 题目：Khoor, Zruog! 解法： 偏移3位（K→H, h→e等），明文为 Hello, World!

iodj{Khoor, Zruog!}

图片

JPEG (jpg):
文件头：FF D8 FF
文件尾：FF D9
PNG (png):
文件头：89 50 4E 47
文件尾：AE 42 60 82

6ced0acc0da4c328dc0349867019d463