

# Misc (杂项) 方向入门指北

by ZeroAurora & littlebean @ XDSEC

## 何谓 Misc?

Misc (Miscellaneous), 意为“杂项”。在 CTF 的概念中, Misc 代表着一切有安全攻防价值, 但内容量并不足以单独组成一个方向的题目。

在国外的各大比赛中, 所谓 Misc 往往是较为细化地分为几类, 具体包括 Recon (信息搜集)、Encoding (编码)、Stego (隐写)、Forensics (取证)、Traffic (流量分析) 等等, 但在国内, 由于比赛规模所限, 这类题目往往合并为一个大类: Misc。

## Misc 简单吗?

如果你一定要我给出一个答案, 我会说: 是, 但不是。

可不要嫌弃我谜语人。一来, 相比于其他方向, Misc 所需的知识深度不深, 但广度要求极高, 且由于近年来计算机科学的热点较多, Misc 的题目往往与时俱进地走在最前头, 例如 AI 攻防、区块链等; 二来, Misc 的难度跨度极大, 经常会出现两极分化的情况。例如许多比赛的 Misc 分区总有那么两题, 一题是整场比赛最简单的签到题, 而另一题则是完全没有一队做出。

综上所述, Misc 虽然入门简单, 但也是一个深坑。不过, 虽然丑话说在了前头, 我依然希望所有对 CTF 感兴趣的朋友们都来试一试这个方向。

## Misc 适合我吗?

“适合”是一个很玄学的话题, 每个人的喜好和胃口都不一样, 我并不能下定论。不过我以为, 如果你喜欢钻研、热爱探索、希望拓展自己的知识面的话, 那么你一定享受学习 Misc 的过程。

## 我该怎么开始学习呢?

首先是一些所有方向, 乃至整个计算机学习都老生常谈的话题: 自学能力。

Search the fxcking web. Read the fxcking manual. 这些都是高效自学的基本方法。当然, 现在是 2023 年, 是生成式 AI 大年, 我想, 还应该加上一个 Ask the fxcking AI。如果你对某些东西有疑惑的话, 不妨先把手头的这些资源都利用起来。

当然, 如果你始终没有找到自己想要的答案的话, 也请不要犹豫, 抱起你的问题和思考过程, 寻找同辈或前辈们的帮助。我想补充的一点是, 在询问他人的过程中, 不要低声下气, 不要过分“自谦”。过度的谦虚就是骄傲, 不分场合地“抱大腿”只会让交流的效率越发降低。

然后就是一些 Misc 特定的学习路线建议啦。

# 环境

Misc 没有环境要求——或者说，Misc 的环境要求没有意义。“杂项”一词的本义注定了这个方向所需工具的多样性乃至生僻性。因此，Misc 题目往往没有固定的环境要求。虽然这么说，但是仍然有几个你一定需要准备的东西：

1. 一个可用的 Linux 操作系统。如果你第一次听说 Linux 这个东西，我建议首先安装几个知名度的 Linux 发行版尝试一下，例如 Debian、Ubuntu、Fedora 等等，也可以使用国内开发者主导的 Deepin 等等。如果你已经比较熟悉 Linux 的使用，那我会推荐使用 Kali Linux，因为它的包管理器的仓库足够大，能够提供绝大部分你需要的工具。安装在何处是无所谓的：我把它安装在 VirtualBox 虚拟机中，而你也许会选择 WSL，也许会选择实机安装，这完全取决于你。
2. 基础的编程语言与工具：尤其是 Python。也许 Python 并不是多好的语言，但是它一向是短小脚本的首选。Misc 方向常有需要编写简易脚本的题目，此时若你对 Python 的掌握足够熟练，就能节省下不少时间。此外，为了便捷地在电脑上运行各种不同的软件而又不至于搞砸系统，你也许也需要学习 Docker 等容器化技术。
3. 针对各个小方向和特定题目，安装一些常用的工具。这在后面会介绍到。

## 分类简介

### Recon（信息搜集）

这类型的题也被称为社工（[社会工程学](#)）题，通常会给你一些线索（比如图片或一些文字）让你成为侦探去推理搜证，一步步的根据线索找到 flag，这时候就要用你能想到的一切方法来寻找线索啦。

一些可能会有用的信息搜集技巧和搜索技巧：（摘自 CTF-wiki）

- 公开渠道
- 目标 web 网页、地理位置、相关组织
- 组织结构和人员、个人资料、电话、电子邮件
- 社会公共信息库查询
- Google 基本搜索和挖掘技巧（要学会科学上网
- 地图与街景：Google Map、Google Earth、百度/高德地图、whois 数据库等

### Encoding（编码）

编码（encode）是信息从一种形式或格式转换为另一种形式的过程。相应的，解码（decode）是编码的逆过程。

编码，其可以理解为，采用一种新的载体来表示前一个载体所表达的信息。可以套用类似这样一个公式来理解：XX 编码，将 A 编码为 B，以实现通过 B 进行存储或传输的目的。比如摩斯

**电码**，将“文本数据”编码为“点横组成的电信号”，以实现通过“电报”进行传输的目的。

misc 的题里总是会出现一些奇奇怪怪的乱码和字符，他们很有可能就是 flag 编码后的样子，我们要做的就是熟悉各种编码，并准确的识别出他们，只要能分辨出是哪一种编码，在网上可以搜到相应编码的对照表、在线的转换工具、解码脚本等，然后就可以将 flag 解码还原出来了。

这里给大家介绍几种常见的编码：

## 进制转换

进制也就是进位计数制，是人为定义的带进位的计数方法（有不带进位的计数方法，比如原始的结绳计数法，唱票时常用的“正”字计数法，以及类似的 tally mark 计数）。对于任何一种进制—X 进制，就表示每一位置上的数运算时都是逢X进一位。十进制是逢十进一，十六进制是逢十六进一，二进制就是逢二进一，以此类推，x进制就是逢x进位。

在CTF比赛中，常见进制为二进制、八进制、十进制、十六进制。

常规考点：进制转换，flag 有可能被不同的进制进行编码，当遇到多个进制组合编码后，可利用 python 或其他脚本语言编写脚本进行解码，也可手搓（233333）。

## ASCII编码

**简述：**

ASCII 码是对**英语字符与二进制位**之间的关系，做了统一规定。

基本的 ASCII 字符集共有 128 个字符，其中有 96 个可打印字符，包括常用的字母、数字、标点符号等。

ASCII码是一种用于表示字符的编码系统，它是计算机发展早期最常用的编码系统之一。

**特征：只含有数字**

- 0-9对应的ASCII码是49-57
- A-Z对应的ASCII码是65-90
- a-z对应的ASCII码是97-122

ASCII在线解码、ASCII码对照表在网上是很容易搜到的

## base 家族

base 家族的成员有很多，比如：base16 base32 base64 base85 base36 base 58 base91 base 92 base62...

base xx 中的 xx 表示的是采用多少个字符进行编码，比如说 base64 就是采用 64 个字符进行编码。

**其中最常见的是 base64**，他由共64个(A-Z a-z 0-9 + /)可打印字符组成，对应0-63，末尾最多2

个=

末尾有等号是 **base64** 一大特征，但是不是每个base64编码后的结果都有等号，有等号的也不一定是base64

至于为什么会有等号看这里[base 家族的编码原理](#)

## 其他编码

MISC题会出的编码和加密还有很多很多.....

这里有一篇介绍的比较全面的博客 [CTF常见编码及加解密（超全） - ruoli-s - 博客园 \(cnblogs.com\)](#)

那么当我们遇到不认识或者没见过的编码怎么办呢？答案是搜索引擎, 不认识的编码丢进去搜搜试试。

搜到后就是找工具或者自己写脚本解决 —(虽然但是都有在线工具子我为什么还

[CyberChef](#) (最常用)

[Ciphey](#) (在未知编码或加密算法的时候可以用来碰碰运气，比 CyberChef Magic Recipe 更好)

[CTF在线工具-CTF工具|CTF编码|CTF密码学|CTF加解密|程序员工具|在线编解码 \(ssleye.com\)](#)

[在线工具 - Bugku CTF\(在线工具 - Bugku CTF\)\)](#)

## Stego (隐写)

隐写就是“我有小秘密被我藏起来了，你要想办法找到它”。

- **隐写术**是一门关于**信息隐藏**的技巧与科学
- **信息隐藏**指的是不让预期接收者之外的任何人知晓信息的传递或者内容

隐写是 misc 中最经典最常见一部分，一般会把 flag 信息藏在一个的文件里（图片、音频、文档，甚至是压缩包里，一切文件都有可能），这时候我们就要学习一些文件结构、文件的特性等等（这里还要学会使用一些十六进制编辑器），flag 信息往往就藏在这里边。常见的隐写网上搜索就能找到好多，快去快去搜索学习一下！

能用于 misc 的隐写有很多很多，不同的隐写方式也会有不同的工具或脚本使用。所以平时做题一定要学会发挥搜索引擎的最大作用，最重要的是多做题多积累多总结。

## Forensics (取证)

取证听起来就好酷的好不啦，任何邪恶终将（不好意思串台了

电子数据取证是指能够为法庭接受的、足够可靠和有说服力的、存在于计算机和相关外设中的电

子证据的确定、收集、保护、分析、归档以及法庭出示的过程。

CTF 中的取证和现实中的取证不同，现实中的取证很少会涉及巧妙的编码加密，数据隐藏，被分散嵌套在各处的文件字符串，或是其他脑洞类的 Challenge。很多时候是去精心恢复一个残损的文件，挖掘损坏硬盘中的蛛丝马迹，或者从内存镜像中抽取有用的信息。（这段取自 CTFwiki）

CTF 中的电子数据取证主要分为硬盘取证和内存取证两部分，并且考察对证据文件的分析。

## 常用工具

- 硬盘镜像取证工具 FTK Imager AutoPsy
- 内存镜像取证工具 Volatility
- 加密容器工具 Veracrypt
- 各大国内取证技术提供商的工具

更多取证相关[Xidian Forensics | Home](#)

## Traffic（流量分析）

流量分析是指捕捉网络中流动的数据包，并通过查看包内部数据以及进行相关的协议、流量分析、统计 等来发现网络运行过程中出现的问题。

现实中流量分析作用通常是溯源攻击流量的（就是想办法找到发起攻击的大黑阔

在CTF比赛中会拿到一个 .pcap/.pcapng 文件，它是由捕获的网络流量形成的，而考点在于利用流量分析工具，抓取网络请求中的各种流量数据包，分析信息，并得到所需的有用信息（flag）。

复杂的地方在于数据包里充满着大量无关的流量信息，学会如何分类和过滤数据是非常重要的!!!

常用工具：Wireshark

一些可能需要了解的知识

- [OSI 七层模型](#)
- [流量分析入门](#)
- [Wireshark 使用](#)

流量分析和上边提到的取证都是需要比较多的前置知识的，遇到不懂的东西一定要善于使用搜索引擎去了解（搜索引擎对于misc来说真的是必不可少的东西哇!）

## 总结

MISC是一个具有极大趣味性的方向，也是切入CTF 竞赛领域、培养兴趣的一个很不错的入口，希望大家都能找到乐趣，学到知识，结识好朋友。祝大家在 MoeCTF2023 玩的开心啦~

最后是给大家准备的 flag，不过已经被编码过了，看看你能不能解码出来（提示一下，文中就有工具哦）：`bW9lY3Rme2hAdjNfZnVuX0B0X20xNWNfIX0=`