🕵️ Forensic CTF Challenge Write-up

Author: Karthik

Category: Forensics — *Disk Image Analysis (.dd file)*

🧠 Introduction

In this forensic Capture The Flag (CTF) challenge, we are provided with a raw disk image file (`file.dd`). Disk image files in .dd format are often exact bit-by-bit copies of physical storage devices, and they're commonly used in digital forensics investigations. My task was to analyze this image, explore its contents, and extract a hidden flag.

The challenge simulates a real-world forensic workflow—starting with basic string analysis and ending with intelligent keyword filtering. This write-up details my methodology, tools used, challenges faced, and how I recovered the flag.

🧰 Tools Used

- Linux command line
- `strings`: Extract printable strings from binary files
- `grep`: Search for specific patterns in output
- `file` (optional): To verify file types
- CTF intuition and pattern recognition

🧪 Step-by-Step Approach

1. Understanding the Disk Image File (`file.dd`)

The `.dd` file is a raw image of a disk, containing all data from a storage device—including deleted content, metadata, and unallocated space. Valuable data such as passwords or flags can often be found here.

2. Initial String Extraction

First, I used the `strings` command to extract all printable characters from the binary file:

```
strings disk.dd
```

This produced a large amount of output. While most of it was noise, some of it could potentially include flags or clues.

3. Using Pattern Matching to Narrow Down Results

Since CTF flags often follow a predictable format like `CTF{}`, `FLAG{}`, or in this case `PIRATE{}`, I used `grep` to filter strings:

```
strings disk.dd | grep PIRATE
```

This quickly narrowed down results and exposed the hidden flag:

PIRATE{skull_n_bones}

🏁 Final Flag

PIRATE{skull_n_bones}

(Replace the flag with the actual one if different.)

🔍 Conclusion

This challenge demonstrates how powerful simple command-line tools can be in digital forensics. No need for complex software—just a smart strategy and basic Linux utilities.

In real-world investigations, this type of quick triage can reveal valuable evidence with minimal resources. It's a reminder that knowing your tools is just as important as the tools themselves.

🧩 Takeaways

- Raw disk images can contain hidden data—even in unallocated space.
- The `strings` command is essential for extracting readable content from binary files.
- Pattern matching with `grep` increases efficiency in data filtering.
- Simple tools + solid intuition = powerful forensic results.