**CTF Solutions Preparation**

**1. Organize Your Solutions**

**Make sure your solutions are neatly organized into directories, with clear names and descriptions. For each challenge, you should have:**

* **Solution Files: Scripts, binaries, or any other files required to solve the challenge.**
* **Documentation: A brief explanation of how you approached the problem and the solution.**

**2. Documentation**

**Create a README.md file to document your solutions. Here’s a template to guide you:**

**File Name: README.md**

**markdown**

**CTF Solutions**

**Overview**

**This repository contains solutions for the Capture The Flag (CTF) competition [Name of CTF], held on [Date]. Each directory corresponds to a challenge from the competition.**

**Challenges**

**1. Challenge 1: [Challenge Name]**

**- Description: Provide a brief description of the challenge.**

**- Solution: Explain the approach you took to solve the challenge.**

**- Files: List the files related to this challenge, e.g., `challenge1.py`, `exploit.sh`.**

**2. Challenge 2: [Challenge Name]**

**- Description: Provide a brief description of the challenge.**

**- Solution: Explain the approach you took to solve the challenge.**

**- Files: List the files related to this challenge, e.g., `challenge2.txt`, `solution.c`.**

**Instructions**

**How to Run**

**Provide clear instructions on how to run or use the solution files. For example:**

**1. Navigate to the challenge directory: `cd challenge1`**

**2. Run the script: `python challenge1.py`**

**Dependencies**

**List any dependencies or libraries required to run the solutions. For example:**

**- Python 3.x**

**- `requests` library (install with `pip install requests`)**

**Contact**

**If you have any questions or need further clarification, please contact me at [Your Email].**

**3. Example Solutions**

**Here’s an example of how to structure and document solutions for a hypothetical CTF competition:**

**File Structure:**

**ctf-solutions/**

**├── challenge1/**

**│ ├── challenge1.py**

**│ └── README.md**

**├── challenge2/**

**│ ├── exploit.sh**

**│ └── README.md**

**└── README.md**

**Directory: challenge1**

**File: challenge1/README.md**

**markdown**

**Challenge 1: Phishing Email Classifier**

**Description**

**This challenge required developing a tool to classify phishing emails. The solution involves building a text classification model.**

**Solution**

**The solution uses a Naive Bayes classifier to differentiate between phishing and non-phishing emails.**

**Files**

**- `challenge1.py`: Python script implementing the phishing detection tool.**

**Instructions**

**How to Run**

**1. Ensure Python 3.x is installed.**

**2. Install the required libraries: `pip install scikit-learn`.**

**3. Run the script: `python challenge1.py`**

**Dependencies**

**- Python 3.x**

**- `scikit-learn` library (install with `pip install scikit-learn`)**

**Contact**

**For any questions, contact [Your Email].**

**Directory: challenge2**

**File: exploit.sh**