Struttura COFELET

- **Scenario**: Un programma C che implementa un buffer overflow vulnerabile utilizzando `gets()`, permettendo agli utenti di manipolare l'esecuzione del programma per ottenere una flag.
- **Task**: Utilizzare un attacco di buffer overflow per chiamare la funzione `wish` con gli argomenti corretti e ottenere la flag nascosta.
- **Goal**: Il giocatore deve scoprire il buffer overflow e manipolare il flusso di esecuzione per ottenere la flag nascosta nel programma. L'obiettivo finale è chiamare correttamente la funzione `wish` passando i valori "Appari" e "Shenron".
- **Condition**: Il giocatore ha accesso al binario vulnerabile, ma non può modificare il codice sorgente. Deve utilizzare tecniche di overflow e ROP (Return Oriented Programming) per raggiungere l'obiettivo.
- **Scenario Execution Flow (SEF)**:
    1. Scoprire che la funzione `gets()` è vulnerabile a un attacco di buffer overflow.
    2. Identificare la funzione `wish` e i valori che deve ricevere per restituire la flag.
    3. Usare strumenti di reversing e debugging per trovare i gadget necessari per manipolare i registri RDI e RSI.
    4. Costruire un payload che sfrutti il buffer overflow e manipoli i registri corretti per chiamare la funzione `wish`.
- **Learning Context**: Il giocatore imparerà a identificare e sfruttare vulnerabilità di buffer overflow, nonché a manipolare il flusso di esecuzione utilizzando gadget ROP.
- **Teaching Content**: Esempi di vulnerabilità di overflow con focus su funzioni non sicure come `gets()`. Introduzione a tecniche ROP per manipolare i registri e chiamare funzioni con parametri.