Struttura COFELET

1. **Task**:
    - Eseguire un attacco di **Command Injection** bypassando i controlli di input per leggere il contenuto del file `flag.txt`.
2. **Goal**:
    - L'obiettivo è sfruttare la vulnerabilità di Command Injection per ottenere la flag leggendo il file `flag.txt`.
3. **Condition**:
    - L'utente deve conoscere le basi dell'iniezione di comandi e delle tecniche di evasione per bypassare i controlli di input, come l'uso di `${IFS}` e altri metodi.
4. **Scenario Execution Flow (SEF)**:
    - **Step 1**: Identificare l'applicazione vulnerabile all'iniezione di comandi tramite il parametro `hostname`.
    - **Step 2**: Preparare un payload per bypassare i controlli di input.
    - **Step 3**: Eseguire il payload e visualizzare il contenuto del file `flag.txt` usando `cat`.
5. **Knowledge, Skills, and Abilities (KSA)**:
    - **Knowledge**: Comprensione delle vulnerabilità di Command Injection.
    - **Skills**: Capacità di aggirare i controlli di input usando tecniche di evasione.
    - **Abilities**: Identificare punti vulnerabili in un'applicazione e sfruttare il sistema operativo sottostante.
6. **Learning Objects**:
    - Alla fine dell'esperienza, i partecipanti dovranno aver imparato come individuare una vulnerabilità di Command Injection, come sfruttare la shell del sistema operativo tramite l'injection, e come aggirare i controlli insufficienti sull'input dell'utente.
7. **Scenario**:
    - L'utente gioca il ruolo di un attaccante che tenta di infiltrarsi in un sistema sfruttando un'input non correttamente filtrato per eseguire comandi arbitrari, con lo scopo di ottenere l'accesso a file riservati come `flag.txt`.