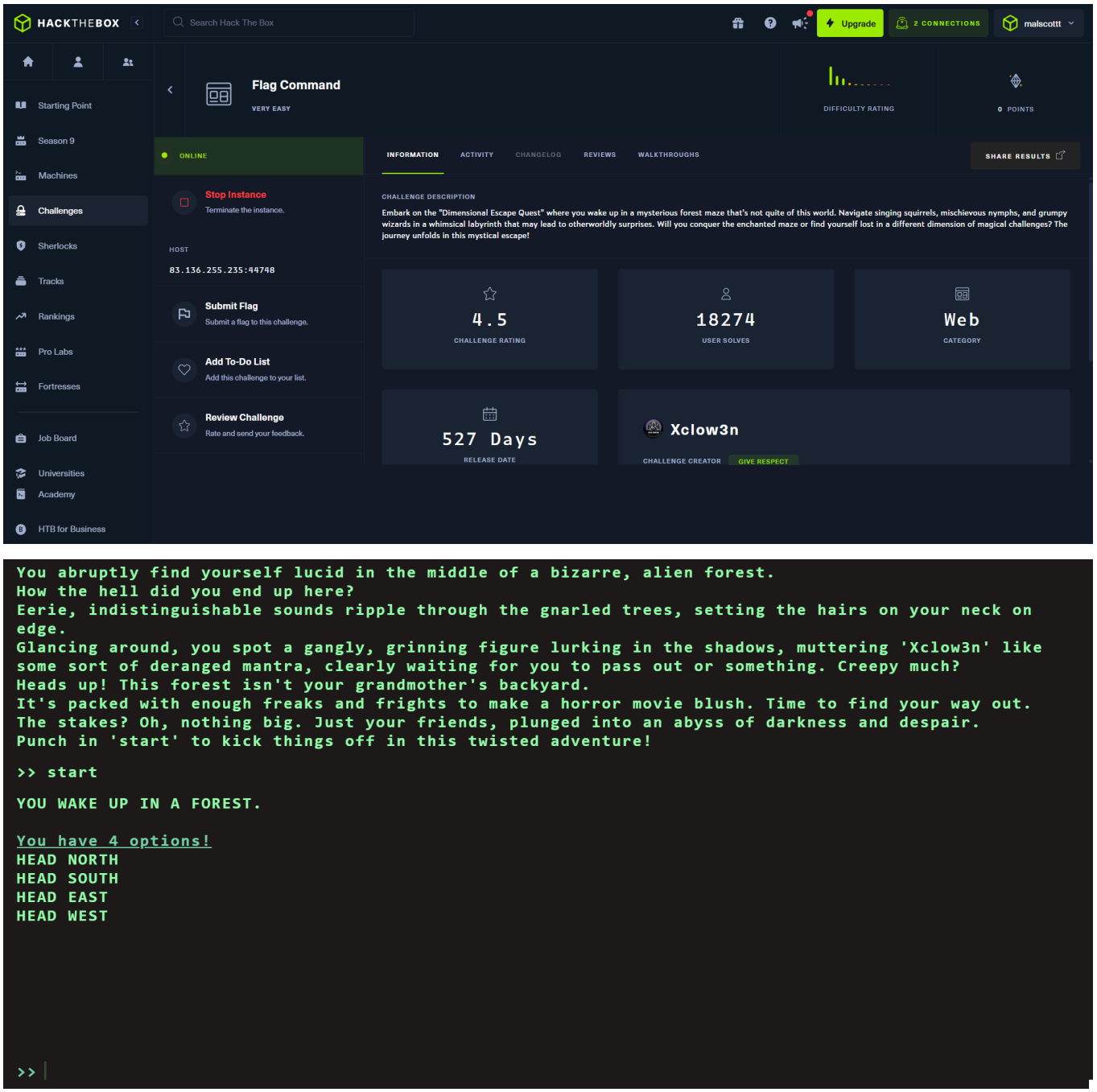
flag command (very easy)




```
You abruptly find yourself lucid in the middle of a bizarre, alien forest.
How the hell did you end up here?
Eerie, indistinguishable sounds ripple through the gnarled trees, setting the hairs on your neck on
edge.
Glancing around, you spot a gangly, grinning figure lurking in the shadows, muttering 'Xclow3n' like
some sort of deranged mantra, clearly waiting for you to pass out or something. Creepy much?
Heads up! This forest isn't your grandmother's backyard.
It's packed with enough freaks and frights to make a horror movie blush. Time to find your way out.
The stakes? Oh, nothing big. Just your friends, plunged into an abyss of darkness and despair.
Punch in 'start' to kick things off in this twisted adventure!

>> start

YOU WAKE UP IN A FOREST.

You have 4 options!
HEAD NORTH
HEAD SOUTH
HEAD EAST
HEAD WEST

>>
```

first, i viewed the page source from the main page.

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Flag Command</title>
  <link rel="stylesheet" href="/static/terminal/css/terminal.css" />
  <link rel="stylesheet" href="/static/terminal/css/commands.css" />
</head>

<body style="color: #94ffaa !important; position: fixed; height: 100vh; overflow: scroll; font-size: 28px;font-weight: 700;">
  <div id="terminal-container" style="overflow: auto;height: 90%;">
    <a id="before-div"></a>
  </div>
  <div id="command">
    <textarea id="user-text-input" autofocus></textarea>
    <div id="current-command-line">
      <span id="commad-written-text"></span><b id="cursor">&#9608;</b>
    </div>
  </div>
  <audio id="typing-sound" src="/static/terminal/audio/typing_sound.mp3" preload="auto"></audio>
  <script src="/static/terminal/js/commands.js" type="module"></script>
  <script src="/static/terminal/js/main.js" type="module"></script>
  <script src="/static/terminal/js/game.js" type="module"></script>

  <script type="module">
    import { startCommander, enterKey, userTextInput } from "/static/terminal/js/main.js";
    startCommander();

    window.addEventListener("keyup", enterKey);

    // event listener for clicking on the terminal
    document.addEventListener("click", function () {
      userTextInput.focus();
    });


  </script>
</body>

</html>
```

as we can see, there's another HTML `<script>` tags.

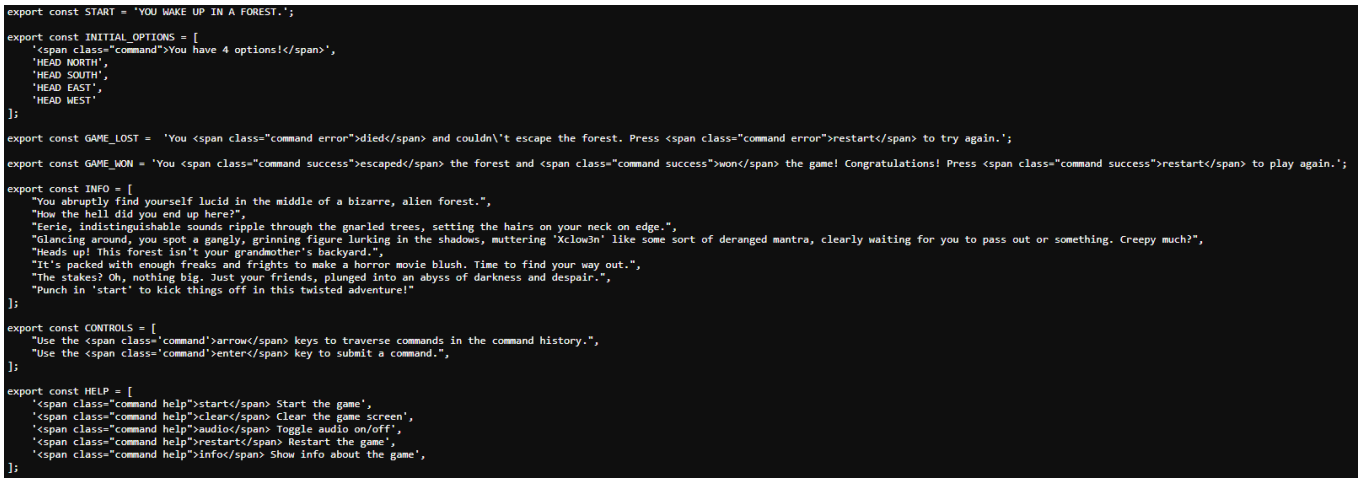
```
    <script src="/static/terminal/js/commands.js" type="module"></script>
    <script src="/static/terminal/js/main.js" type="module"></script>
    <script src="/static/terminal/js/game.js" type="module"></script>
```

- **http[:]//83[.]136[.]255[.]235[:]44748/static/terminal/js/commands.js**

```javascript
import { displayLineInTerminal } from "./main.js";
import { GAME_LOST, GAME_WON } from "./commands.js";

// GAME MECHANICS
// ---------------------------------------
const timeDelay = 1000;

function displayGameResult(message, style) {
    setTimeout(() => {
        displayLineInTerminal({
            text: message,
            style: `${style} margin-right`,
            useTypingEffect: true,
            addPadding: true,
        });
    }, timeDelay);
}

export function playerLost() {
    displayGameResult(GAME_LOST, "error");
}

export function playerWon() {
    displayGameResult(GAME_WON, "success");
}
```

this is the javascript code from `commands.js` file. the entire purposes is to manage the game's end state which is winning or losing.

- **http[:]//83[.]136[.]255[.]235[:]44748/static/terminal/js/game.js**



```
export const START = 'YOU WAKE UP IN A FOREST.';

export const INITIAL_OPTIONS = [
    '<span class="command">You have 4 options!</span>',
    'HEAD NORTH',
    'HEAD SOUTH',
    'HEAD EAST',
    'HEAD WEST'
];

export const GAME_LOST =  'You <span class="command error">died</span> and couldn\'t escape the forest. Press <span class="command error">restart</span> to try again.';

export const GAME_WON = 'You <span class="command success">escaped</span> the forest and <span class="command success">won</span> the game! Congratulations! Press <span class="command success">restart</span> to play again.';

export const INFO = [
    "You abruptly find yourself lucid in the middle of a bizarre, alien forest.",
    "How the hell did you end up here?",
    "Eerie, indistinguishable sounds ripple through the gnarled trees, setting the hairs on your neck on edge.",
    "Glancing around, you spot a gangly, grinning figure lurking in the shadows, muttering 'Xclow3n' like some sort of deranged mantra, clearly waiting for you to pass out or something. Creepy much?",
    "Heads up! This forest isn't your grandmother's backyard.",
    "It's packed with enough freaks and frights to make a horror movie blush. Time to find your way out.",
    "The stakes? Oh, nothing big. Just your friends, plunged into an abyss of darkness and despair.",
    "Punch in 'start' to kick things off in this twisted adventure!"
];

export const CONTROLS = [
    "Use the <span class='command'>arrow</span> keys to traverse commands in the command history.",
    "Use the <span class='command'>enter</span> key to submit a command.",
];

export const HELP = [
    '<span class="command help">start</span> Start the game',
    '<span class="command help">clear</span> Clear the game screen',
    '<span class="command help">audio</span> Toggle audio on/off',
    '<span class="command help">restart</span> Restart the game',
    '<span class="command help">info</span> Show info about the game',
];
```

this is the javascript code from `game.js`. the purposes is to define and export all the text that will need to display in the game.

- **http[:]//83[.]136[.]255[.]235[:]44748/static/terminal/js/main.js**

```
async function CheckMessage() {
    fetchingResponse = true;
    currentCommand = commandHistory[commandHistory.length - 1];

    if (availableOptions[currentStep].includes(currentCommand) ||
availableOptions['secret'].includes(currentCommand)) {
        await fetch('/api/monitor', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json'
            },
            body: JSON.stringify({ 'command': currentCommand })
        })
```

this snippet shows it has a secret command to reveal the flag in `availableOptions['secret']`. which means we must lurking for the secret word or command to reveal the flag.

```
const fetchOptions = () => {
    fetch('/api/options')
        .then((data) => data.json())
        .then((res) => {
            availableOptions = res.allPossibleCommands;

        })
        .catch(() => {
            availableOptions = undefined;
        })
}
```

from this snippet, the vulnerability from this flag is from `/api/options`. this code above fetches a JSON file from `/api/options` and stores all possible commands in the `availableOptions`.

- **http[:]//83[.]136[.]255[.]235[:]44748/api/options**

```
{
  "allPossibleCommands": {
    "1": [
      "HEAD NORTH",
      "HEAD WEST",
      "HEAD EAST",
      "HEAD SOUTH"
    ],
    "2": [
      "GO DEEPER INTO THE FOREST",
      "FOLLOW A MYSTERIOUS PATH",
      "CLIMB A TREE",
      "TURN BACK"
    ],
    "3": [
      "EXPLORE A CAVE",
      "CROSS A RICKETY BRIDGE",
      "FOLLOW A GLOWING BUTTERFLY",
      "SET UP CAMP"
    ],
    "4": [
      "ENTER A MAGICAL PORTAL",
      "SWIM ACROSS A MYSTERIOUS LAKE",
      "FOLLOW A SINGING SQUIRREL",
      "BUILD A RAFT AND SAIL DOWNSTREAM"
    ],
    "secret": [
      "Blip-blop, in a pickle with a hiccup! Shmiggity-shmack"
    ]
  }
}
```


from the url, we got the secret word in `/api/options`.

```
    "secret": [
      "Blip-blop, in a pickle with a hiccup! Shmiggity-shmack"
    ]
```

we paste the secret word to the game to print the flag.

```
You abruptly find yourself lucid in the middle of a bizarre, alien forest.
How the hell did you end up here?
Eerie, indistinguishable sounds ripple through the gnarled trees, setting the hairs on your neck on edge.
Glancing around, you spot a gangly, grinning figure lurking in the shadows, muttering 'Xclow3n' like some sort of deranged mantra, clearly waiting for you to pass out or something. Creepy much?
Heads up! This forest isn't your grandmother's backyard.
It's packed with enough freaks and frights to make a horror movie blush. Time to find your way out.
The stakes? Oh, nothing big. Just your friends, plunged into an abyss of darkness and despair.
Punch in 'start' to kick things off in this twisted adventure!

>> start

YOU WAKE UP IN A FOREST.

You have 4 options!
HEAD NORTH
HEAD SOUTH
HEAD EAST
HEAD WEST

>> Blip-blop, in a pickle with a hiccup! Shmiggity-shmack

HTB{D3v3l0p3r_t00l5_4r3_b35t__t0015_wh4t_d0_y0u_Th1nk??}

You escaped the forest and won the game! Congratulations! Press restart to play again.
```

`HTB{D3v3l0p3r_t00l5_4r3_b35t__t0015_wh4t_d0_y0u_Th1nk??}`

from this challenge, this is known as **Broken Access Control** caused by **Sensitive Information Exposure** and **Client-Side Enforcement of Security**. in this challenge, the server should never send a secret or confidential information to the client which is in `/api/options`.