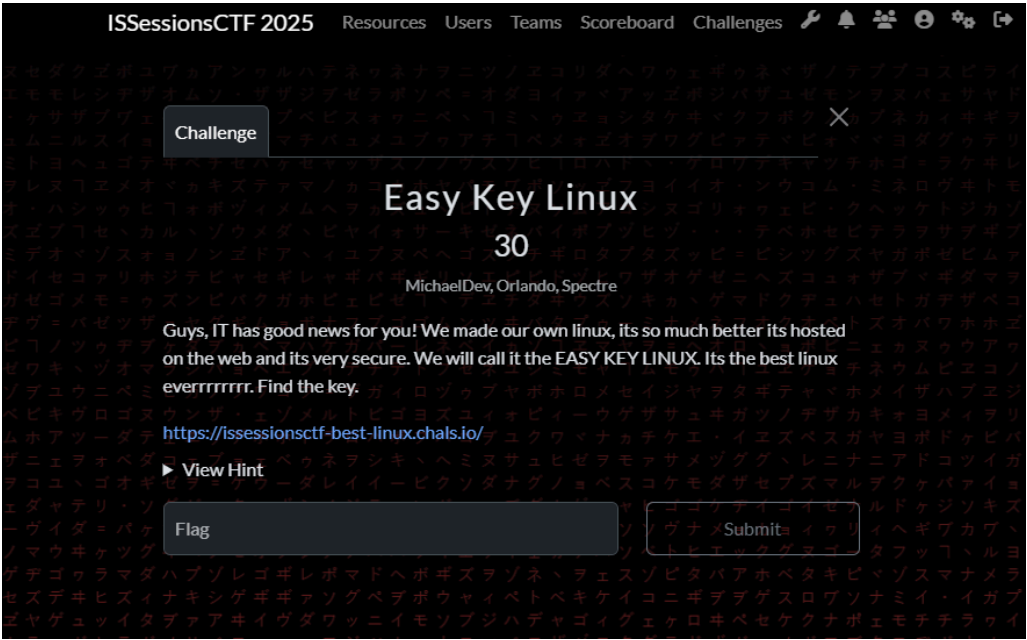


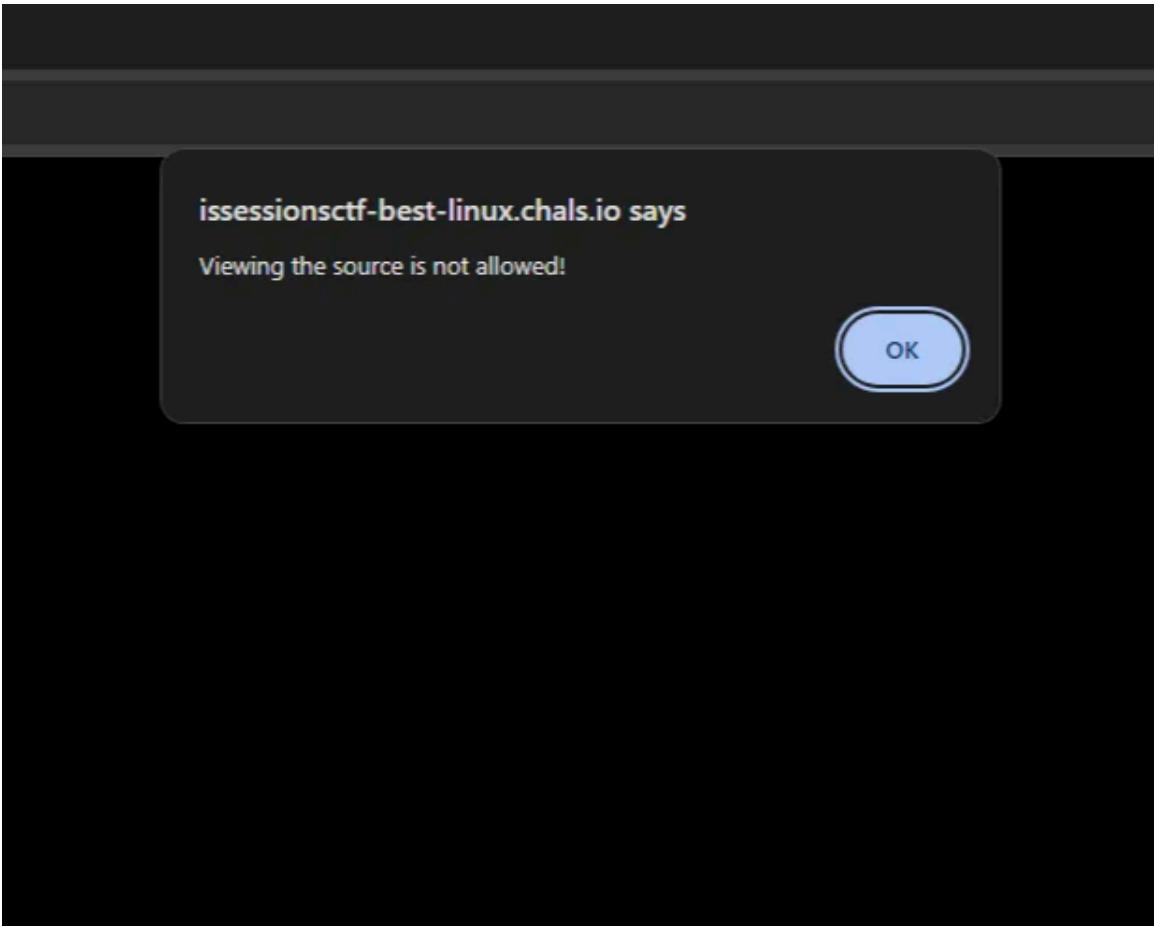
Easy Key Linux (Orlando,Michael, Prajwal)



This is a simple challenge for those who were struggling with the harder more complex challenges. The whole idea is that a developer decided to make their own version of linux which works similar to linux but not exactly the same.

Solution 1

There are actually 2 solutions for this challenge. One of them is looking at the source code and the flag would be present. To put a road block the player can't 'right click' or use 'control u'.



There are multiple ways of bypassing this but one of them is just keep pressing 'control u' or keep pressing 'right click'.

Intitally the player doesn't see much.

```
<!DOCTYPE html>
<html>
<head>
  <title>Linux Terminal PDF</title>
  <style>
    body { background-color: black; color: white; font-family: monospace; padding: 20px; }
    #output { white-space: pre-wrap; }
    input { background: black; color: white; border: none; outline: none; width: 100%; }
    html { display: none; }
    /* Useless CSS for obfuscation */
    .hiddenClass1 { color: red; font-size: 10px; }
    .hiddenClass2 { display: flex; justify-content: center; }
    .hiddenClass3 { transform: rotate(45deg); }
    .hiddenClass4 { opacity: 0.5; }
    .hiddenClass5 { position: absolute; top: -9999px; }
  </style>
  <script>
    document.addEventListener("DOMContentLoaded", function() {
      document.documentElement.style.display = "block";
    });
    document.addEventListener("keydown", function(e) {
      if (e.ctrlKey && (e.key === "u" || e.key === "U" || e.key === "s" || e.key === "i" || e.key === "j" || e.key === "p" || e.key === "o")) {
        alert("Viewing the source is not allowed!");
        e.preventDefault();
      }
    });
    document.addEventListener("contextmenu", function(e) {
      alert("Right-click is disabled!");
      e.preventDefault();
    });
    document.addEventListener("keydown", function(e) {
      if ((e.ctrlKey && e.shiftKey && e.key === "I") || (e.ctrlKey && e.shiftKey && e.key === "J") || (e.ctrlKey && e.key === "S") || (e.key === "F12") || (e.ctrlKey && e.key === "P") || (e.ctrlKey && e.key === "O")) {
        alert("Inspecting elements is disabled!");
        e.preventDefault();
      }
    });
  </script>
</head>
```

If the player scrolls down to the very bottom they will find the following:

```
<body>
  <div id="output">This is our own Linux, it may or may not work like Linuxâ€™who knows? We are cool like that.</div>
  <input type="text" id="command" autofocus>

  <script>
    var fsData = {
      "files": ["file1.txt", "file2.txt"],
      "hidden": { "hidden": "I like cats but not this cat" }
    };
    var rootFs = {
      "files": ["rootfile1.txt", "rootfile2.txt"],
      "hidden": { "hidden": "It may be more hidden, but as long as you know what you are looking for...", "flag": "RootFlag{SuperUser_Access_Granted}" }
    };
    var isRoot = false;

    function promptDisplay() {
      return isRoot ? "root@linux:# " : "user@linux:$ ";
    }

    document.getElementById("command").addEventListener("keydown", function(event) {
      if (event.key === "Enter") {
        let input = this.value.trim();
        let outputDiv = document.getElementById("output");
        outputDiv.innerHTML += "\n" + promptDisplay() + input;

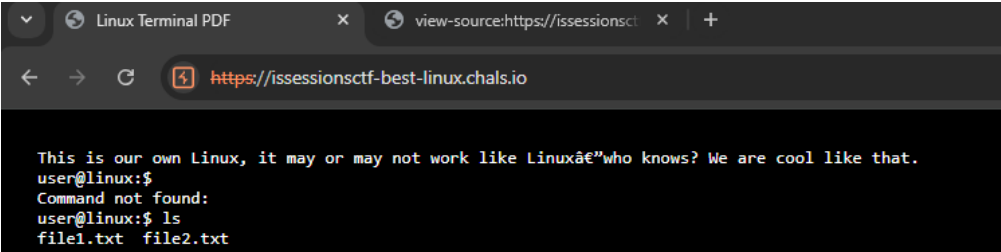
        if (input === "ls") {
          outputDiv.innerHTML += "\n" + (isRoot ? rootFs.files.join(" ") : fsData.files.join(" "));
        } else if (input === "ls -la") {
          outputDiv.innerHTML += "\n.  .. .hidden  " + (isRoot ? rootFs.files.join(" ") : fsData.files.join(" "));
        } else if (input.startsWith("cat ")) {
          let file = input.split(" ")[1];
          if (isRoot && rootFs.hidden[file]) {
            outputDiv.innerHTML += "\n" + rootFs.hidden[file];
          } else if (!isRoot && fsData.hidden[file]) {
            outputDiv.innerHTML += "\n" + fsData.hidden[file];
          } else {
            outputDiv.innerHTML += "\nFile not found: " + file;
          }
        } else if (input === "sudo su -") {
          let password = prompt("Enter password:");
          if (password === "password") {
            isRoot = true;
            outputDiv.innerHTML += "\nSwitched to root user. Type 'ls' to see root files.";
          } else {
            outputDiv.innerHTML += "\nIncorrect password.";
          }
        } else if (input === "exit") {
          isRoot = false;
          outputDiv.innerHTML += "\nExited root mode. Returning to user.";
        } else {
          outputDiv.innerHTML += "\nCommand not found: " + input;
        }

        this.value = ""; // Clear input field
      }
    });
  </script>
</body>
</html>
```

Lastly there was a little trick here which is that the flag is in another format. All the user has to do is put it in the correct format. Believe it or not multiple people fell for this simple little trick.

Solution 2

The first thing most people do when they try to access the page is to use the command ls.



The logical next step is to try and open these files.

```
user@linux:$ cat file1.txt
File not found: file1.txt
user@linux:$ cat file2.txt
File not found: file2.txt
```

These weird linux is for some reason not finding these files. The next step is to see if maybe there are some hidden files.

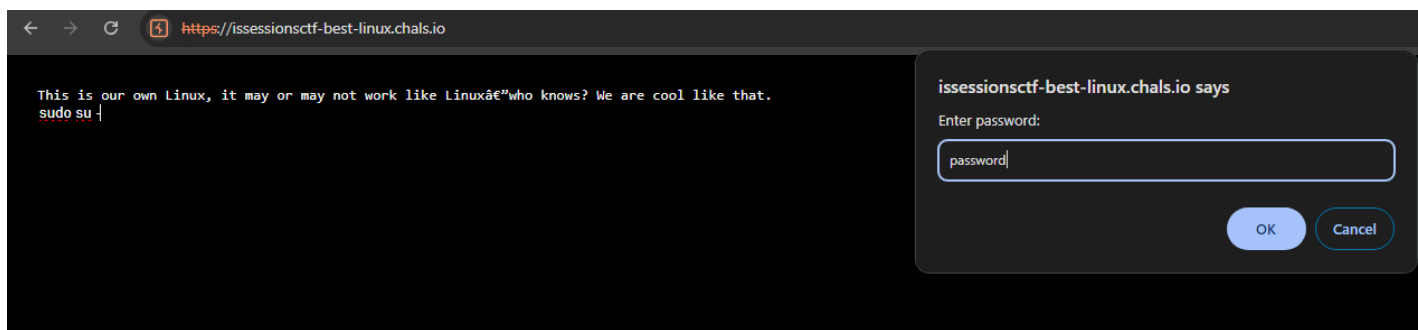
```
user@linux:$ ls -la
.  ..  .hidden  file1.txt  file2.txt
```

There is a hidden file but again it cannot be opened. After testing and testing you try the actual file name without the . because all Easy Key challenges have the solution 'hidden in plain sight'.

```
user@linux:$ cat .hidden
File not found: .hidden
user@linux:$ cat hidden
I like cats but not this cat
```

There is nothing else we can do because cd would also not work. lets try switch to root?

A password is required for the root user. The plan here is for the player to try common passwords used such as root, password



The password 'password' worked and now we are root.

Repeat the same process as before by using ls.

```
user@linux:$ sudo su -
Switched to root user. Type 'ls' to see root files.
root@linux:# ls
rootfile1.txt  rootfile2.txt
```

```
root@linux:# cat rootfile1.txt
File not found: rootfile1.txt
root@linux:# cat rootfile2.txt
File not found: rootfile2.txt
root@linux:# ?
```

Time to check for any hidden files. Since there was one the file can be accessed and it gives the hint to make sure you know what you are looking for which in this case is the flag.

After using the command 'cat flag' then you get the full flag with the incorrect format.

```
root@linux:# ls -la
.  ..  .hidden  rootfile1.txt  rootfile2.txt
root@linux:# cat hidden
It may be more hidden, but as long as you know what you are looking for...
root@linux:# cat flag
RootFlag{SuperUser_Access_Granted}
|
```