JavaScript原型链污染

梅子酒m3i / 2018-09-13 02:06:24 / 浏览数 3984 技术文章 技术文章 顶(1) 踩(0)

博客原文地址:

https://otakekumi.github.io/2018/09/11/JavaScript-%E5%8E%9F%E5%9E%8B%E9%93%BE%E5%8F%8A%E5%8F%98%E9%87%8F%E8%A6%86%E7%9B%96/

原型对象及相关基础

在JavaScript中只有一种结构:对象,也就是常说的"万物皆对象"。

而每个实例对象都有一个原型对象,而原型对象则引申出其对应的原型对象,经过一层层的链式调用,就构成了我们常说的"原型链"。

实例对象可以通过`__proto__`访问其原型对象:

```
>  let obj = {};
<· undefined
>  obj.__proto__;
<· {constructor: ƒ, __defineGetter__: ƒ, __defineSetter__: ƒ, hasOwnProperty: ƒ, __lookupGetter__: ƒ, …}
```

而经过不断的调用,最终的原型对象会调用到`null`,这将作为该原型链的最后一个环节,与之对应的,作为终点的`null`自然也是没有原型对象的。比如,我们继续在上面的例子中调

```
>  obj.__proto__.__proto__;
<· null
```

在JavaScript中,可以用几种方式实现继承,常见的继承方式包括:■■■■■、■■■■■■■■、■■■■■,■■■■■。

每种方法对应的原理及具体的实现方式不再赘述,可参见《JavaScript高级程序语言设计》中的继承相关章节。

在Javascript中可以通过`example.a.b`或`example.a["b"]`来对数组进行访问,`example.a`访问的是`example`对象下的a对象的b,而`example.a["b"]`则是访问`example`对

由于对象是无序的,当使用第二种方式访问对象时,只能使用指明下标的方式去访问。

因此我们可以通过`a["__proto__"]`的方式去访问其原型对象。

实例

在Hackit 2018中,有一个nodejs的题目,其中便涉及到了使用原型链继承来进行变量覆盖的攻击手法,源码我已同步至我的Github项目中CTF-Challenge。

其中漏洞点在第64行:

```
matrix[client.row][client.col] = client.data;
```

而`data`是从网页传递的参数,形如:

```
{"row":1,"col":"1","data":"X"}
```

对应的`row`以及`col`值会存放至`matrix`中,而这里则会导致一个原型链污染的隐患。

回到题目本身的逻辑中,获得Flag的要求为:

> 传入的`querytoken`与`user.admintoken`的md5哈希值一样

因此我们可以通过上面发现的原型链的漏洞对`admintoken`进行赋值。那么继续探讨如何对其进行赋值。

`Array`实例继承自`Array.prototype`,因此我们可以通过更改构造函数的原型对象来对所有的`Array`实例进行修改。

那么我们可以通过这个思路来做到变量覆盖。

通过代码我们可以发现`user`为`Array`,`matrix`同样是`Array`,因此我们根据上面的思路,通过对`matrix`进行赋值进而篡改`user.admintoken`的值。

在控制台演示原型链污染如下:

```
>  a=[];
<· []
>  b=[];
<· []
>  b["__proto__"];
```

```
<· [constructor: ƒ, concat: ƒ, find: ƒ, findIndex: ƒ, pop: ƒ, …]
>  b["__proto__"]["admintoken"]="ccda";
<· "ccda"
>  a
<· []
>  a.admintoken
<· "ccda"
```

因此我们传入的payload就可以构造出来了。

上一篇：某cms 1.4.7 分析 下一篇：Punycode安全威胁浅析

1. 0 条回复
   - 动动手指，沙发就是你的了！