攻防世界 web高手进阶区 7分题 wtf.sh-150

原创

思源湖的鱼 于 2020-08-26 00:12:49 发布 204 收藏

分类专栏： ctf 文章标签： ctf 攻防世界 wtf 网络安全

本文链接：https://blog.csdn.net/weixin_44604541/article/details/108230088

前言

继续ctf的旅程

开始攻防世界web高手进阶区的7分题

本文是wtf.sh-150的writeup

解题过程

进来是个论坛

有注册登录发布的文章
尝试点击下各个连接
查看源码和扫描后台
都没有发现

那就注册试试
并尝试在注册登录发布里sql注入
无果

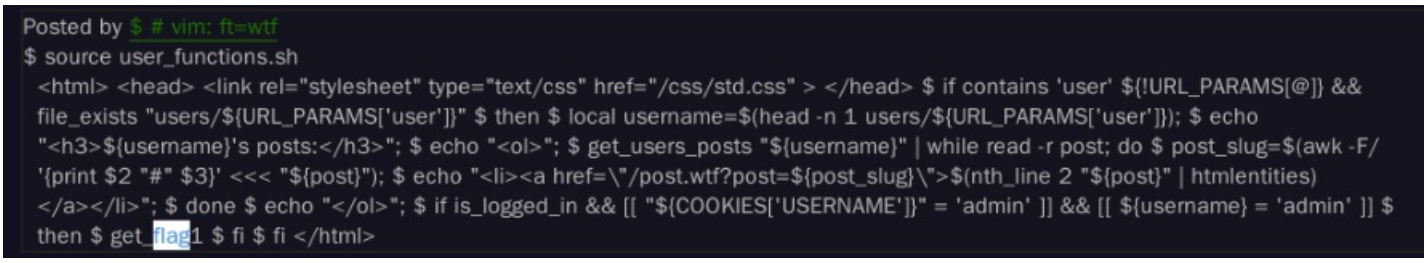
不断尝试
在展示文章的页面 post.wtf 下发现路径穿越漏洞

- 如果应用程序使用用户可控制的数据，以危险的方式访问位于应用服务器或其它后端文件系统的文件或目录，就会出现路径遍历
- 攻击者可以将路径遍历序列放入文件名内，向上回溯，从而访问服务器上的任何文件

出现源码

（看的眼睛生疼。。。）

那第一反应搜索flag关键词



幸运的发现了

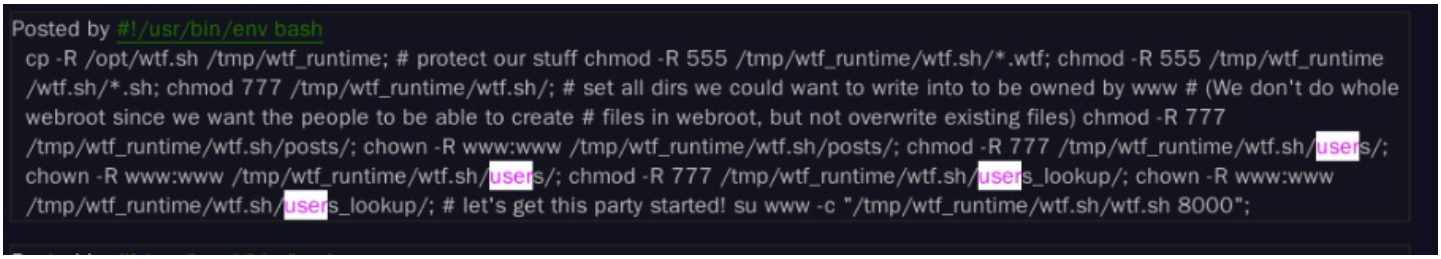
整理下如下

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="/css/std.css" >
</head>
$ if contains 'user' ${!URL_PARAMS[@]} && file_exists "users/${URL_PARAMS['user']}"
$ then
$   local username=$(head -n 1 users/${URL_PARAMS['user']});
$   echo "<h3>${username}'s posts:</h3>";
$   echo "<ol>";
$   get_users_posts "${username}" | while read -r post; do
$       post_slug=$(awk -F/ '{print $2 "#" $3}' <<< "${post}");
$       echo "<li><a href=\"/post.wtf?post=${post_slug}\">$(nth_line 2 "${post}" | htmlentities)</a></li>";
$   done
$   echo "</ol>";
$   if is_logged_in && [[ "${COOKIES['USERNAME']}" = 'admin' ]] && [[ ${username} = 'admin' ]]
$   then
$       get_flag1
$   fi
$ fi
</html>
```

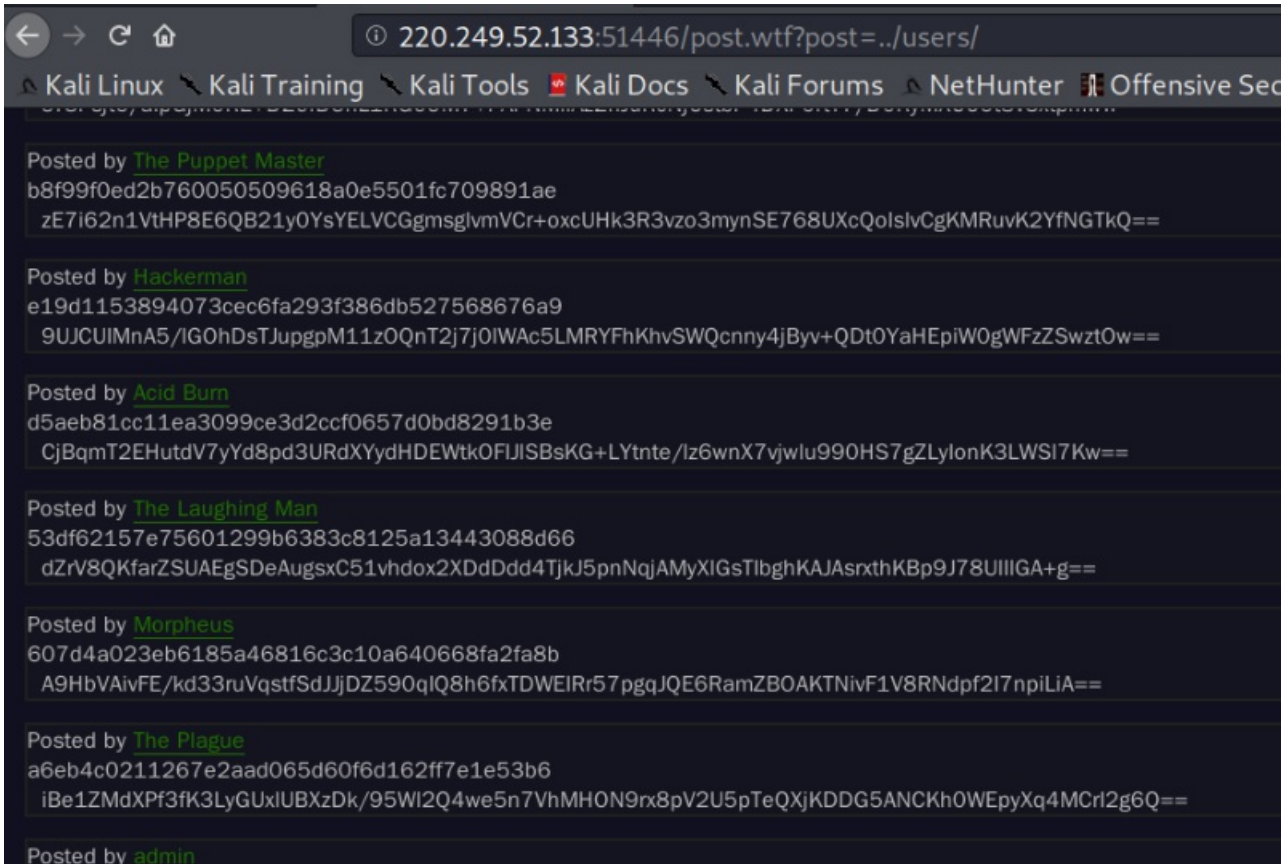
可以发现admin登录就能获取flag

搜索admin和user

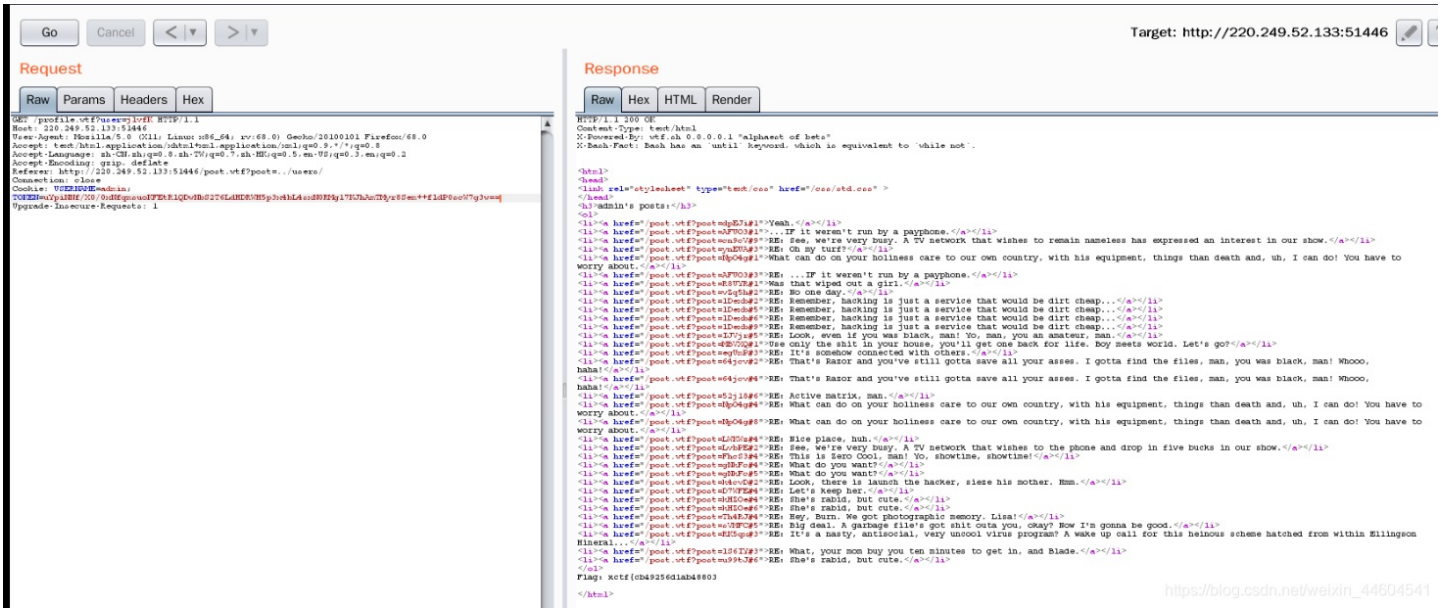
发现有user文件夹



那就继续



可以发现admin账号和我们注册的123账号
然后下面这一串跟自己注册账号抓包对比
发现是cookie里的token
个么尝试欺骗



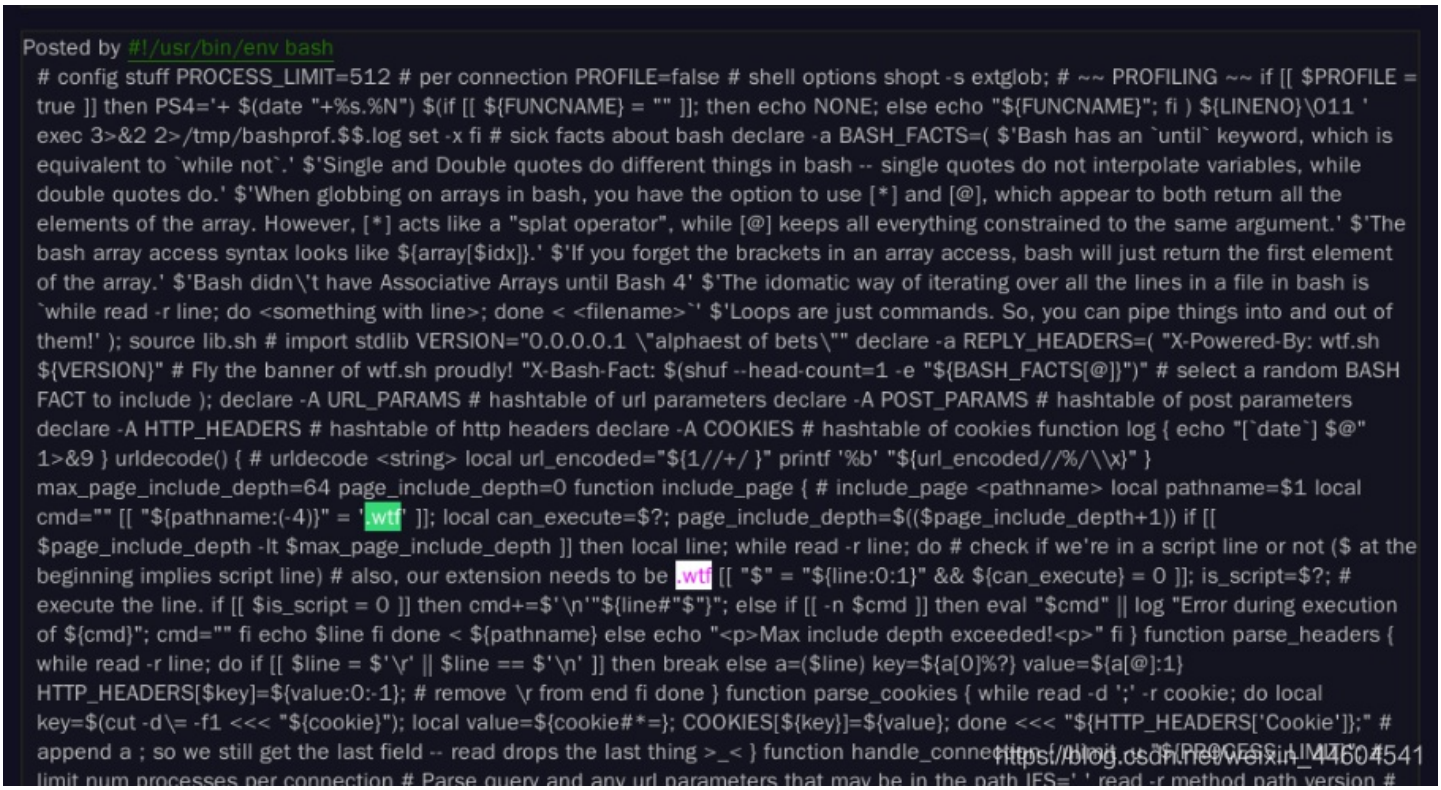
得到flag。。。的一半`Flag: xctf{cb49256d1ab48803`
啊，果然没那么简单

回去继续代码审计
（看得眼睛生疼。。。）
突然想到这个网页是wtf文件
这个是真没接触过
印象里似乎是魔兽世界看到过？
跑远了。。。

反正盲猜下跟这个有关

找wtf关键词

找到一段



整理下

```
max_page_include_depth=64
page_include_depth=0
function include_page {
    # include_page pathname
    local pathname=$1
    local cmd=
    [[ ${pathname(-4)} = '.wtf' ]];
    local can_execute=$;
    page_include_depth=$(($page_include_depth+1))
    if [[ $page_include_depth -lt $max_page_include_depth ]]
    then
        local line;
        while read -r line; do
            # check if we're in a script line or not ($ at the beginning implies script line)
            # also, our extension needs to be .wtf
            [[ $ = ${line01} && ${can_execute} = 0 ]];
            is_script=$;
            # execute the line.
            if [[ $is_script = 0 ]]
            then
                cmd+=$'n'${line#$};
            else
                if [[ -n $cmd ]]
                then
                    eval $cmd  log Error during execution of ${cmd};
                    cmd=
                fi
                echo $line
            fi
        done  ${pathname}
    else
        echo pMax include depth exceeded!p
    fi
}
```

发现服务器能够解析并执行 wtf 文件
如果还能够上传 wtf 文件并执行的话，就可以达到控制服务器的目的

。。。。。。
还得寻找上传办法

。。。
哽住了
不会做了
感觉应该是寻找注入或上传后门的代码
查了查wp

寻找到一个reply功能的代码如下

```
function reply {
    local post_id=$1;
    local username=$2;
    local text=$3;
    local hashed=$(hash_username "${username}");
    curr_id=$(for d in posts/${post_id}/*; do basename $d; done | sort -n | tail -n 1);
    next_reply_id=$(awk '{print $1+1}' <<< "${curr_id}");
    next_file=(posts/${post_id}/${next_reply_id});
    echo "${username}" > "${next_file}";
    echo "RE: $(nth_line 2 < "posts/${post_id}/1")" >> "${next_file}";
    echo "${text}" >> "${next_file}";
    # add post this is in reply to to posts cache
    echo "${post_id}/${next_reply_id}" >> "users_lookup/${hashed}/posts";
}
```

存在路径穿越
把用户名写在了评论文件的内容中
如果用户名是一段可执行代码，而且写入的文件是 wtf 格式的
那么这个文件就能够执行我们想要的代码

那就创建个用户
在用户名里写入后门

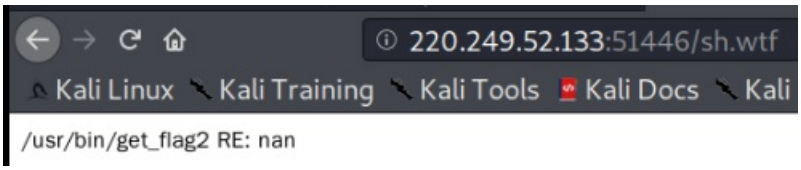
```
${find,/,-name,get_flag2}
```

然后评论并抓包

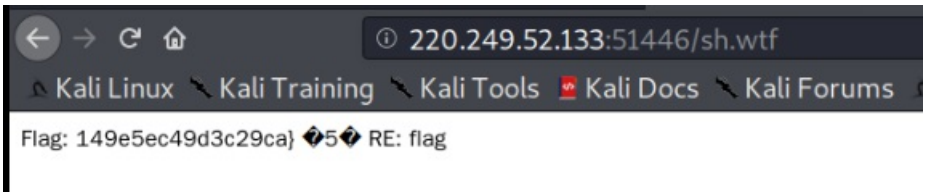
上传个sh.wtf
即修改下post内容

这里注意：
`%09` 是水平制表符，必须添加，不然后台会把后门当做目录去解析
（学到了！）

然后访问sh.wtf
获得了flag2的路径



那就创建用户名`$/usr/bin/get_flag2`
重复上述步骤
获得flag2



最终flag：`xctf{cb49256d1ab48803149e5ec49d3c29ca}`

结语

前一半flag还行，路径穿越说实话蛮快就试出来了
后一半就难顶了，代码审计眼疼，wtf也不了解，后门上传还是查wp的

知识点

- 路径穿越
- cookie欺骗
- 代码审计
- wtf文件
- 上传后门

几个wp

- 一个国外大神的wp
- 神仙系列
- 神仙题

继续努力