

Pwnhub大物必须过Writeup

转载

FLy 鹏程万里 于 2018-06-07 18:53:03 发布 901 收藏

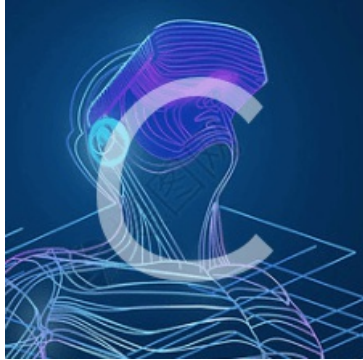
分类专栏: [【CTF】 # 其它Writeup](#)



[【CTF】 同时被 2 个专栏收录](#)

30 篇文章 26 订阅

订阅专栏



[其它Writeup](#)

1 篇文章 0 订阅

订阅专栏

写在最前

这回认真写一次wp，我会尽量写的完整一点，用狼人杀的话说就是把心路历程讲明白。主要两个目的，第一是给做题人一个参考，我在做这题的时候并不知道RPO是啥（看了别人的wp才知道），然而最终还是能做出来，我想这个过程还是值得新人借鉴的，也就是遇到一道新题应该怎么去入手，打CTF遇到自己没见过的知识点太多了，不能期待着撞知识点；第二其实更重要的是给出以后的出题人提一个醒，应该怎么出题。

先总结一下这题，这题我打3分。给3分的原因是纠正了我曾经的一个错误的观点，扣2分的原因有两点，第一点，目标不明确，难以从题目本身正确得出这题想要干什么，第二点，在提交bug页面设置验证码，却不在题目指出该页面的过滤条件，也不指出本题的考点不在该页，浪费做题人时间。下面说说我做题完整过程，供大家参考。

做题

查点（信息收集）

做题的第一步就是应该收集信息，因为题目基本上只有一台服务器，所以直接可上查点，而不是像渗透一样需要先踩点。**做题时的查点三部曲：扫描端口，web目录扫描，浏览网站逻辑。**

端口扫描

通过端口扫描可以收集到服务器运行了什么服务，一是要关注扫描到的开放服务，比如redis，rsync，这时候就要考虑对应服务的攻击，比如未授权访问。二是关注filtered的服务，这往往代表服务器运行着这类服务，但是iptables限制了其访问，比如发现11211是filtered的，而这题恰好有ssrf，就应该考虑用ssrf攻击Memcached服务。

```
Host is up (0.018s latency).
Not shown: 995 closed ports
PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    open       http
445/tcp   filtered   microsoft-ds
3306/tcp  filtered   mysql
4444/tcp  filtered   krb524
```

本题没有开放特别的服务，不过运行着3306，可以推断web是php写的（不绝对）。

web目录扫描

用字典去扫描web目录往往能发现一些出题人留下的提示，比如经典的robots.txt, flag.php, .git, .svn, index.php~, phpinfo.php等等，有时能发现源码泄露，有时能发现隐藏的目录或者文件，或者网站结构。

```
19:01] 200 - 2KB - /index.php3
19:01] 200 - 2KB - /index.php5
19:01] 200 - 2KB - /index.php4
19:01] 200 - 2KB - /index.php~
19:02] 403 - 564B - /lib/flex/uploader/.actionScriptProperties
19:02] 403 - 564B - /lib/flex/uploader/.flexProperties
19:02] 403 - 564B - /lib/flex/varien/.flexLibProperties
19:02] 403 - 564B - /lib/flex/uploader/.project
19:02] 403 - 564B - /lib/flex/uploader/.settings
19:02] 403 - 564B - /lib/flex/varien/.actionScriptProperties
19:02] 403 - 564B - /lib/flex/varien/.project
19:02] 403 - 564B - /lib/flex/varien/.settings
19:02] 200 - 3KB - /login.php
19:03] 200 - 3KB - /login.php
19:04] 200 - 115B - /nginx_status
19:05] 200 - 3KB - /register.php
19:08] 200 - 2KB - /user.php
19:08] 200 - 2KB - /user.php
19:09] 200 - 0B - /router.php
```

本题通过扫描可以发现一个问题，index.phpxxxx都会变成解析index.php，更重要的是，可以发现一个router.php的文件。这意味着该网站应该是通过rewrite将请求统一映射到一个页面，由这个页面路由请求，而这种模式可能遇到%2F问题，对apache或者nginx而言，%2F没有特殊的含义，而对php而言，获得的请求是解码后的，%2F会被解码为/，而/是有含义的，而这个点也是本题的关键（其实我在做到这一步的时候还没有想到%2F的问题，在后来的步骤中想到了）。

浏览网站逻辑

快速浏览一篇网页，并测试网站逻辑，基本可以得出以下几个结论：

- 注册页register.php: Username只允许A-Za-z0-9, Vow测试不存在注入。
- 登录页user.php: Username只允许A-Za-z0-9, 不存在注入。
- 用户页user.php: 不存在越权，只能看自己的用户资料，vow可控，但是被htmlspecialchars编码，在 <p></p> 之间，且强制UTF8编码，不存在xss。
- 课程页classes.php: ID经过intval转换，不存在注入，值得注意的是，存在一个css，使用了../..，而该页本来就在根路径，似乎多此一举。
- 报bug页report_bug.php: Comment只允许A-Za-z0-9, Url必须 http://52.80.19.55/ 开头，不存在SSRF。
- 答题页answer.php: 只要是post就随机返回一个分数，所以该页无用。

分析题目目的，考点

信息收集完，就要开始分析这题的考点，需要我们做什么，是xss，还是ssrf，还是命令注入，然后分析怎么去达到目的。

本题存在一个报告bug的页面，基本可以判断该题为xss，或者是想办法让bot请求用户网站（获取referer or其他的http header），由于不存在xss点，所以这题应该是要诱使机器人访问自己的服务器。

分析完目的，接下来要想办法达到目的，不过到目前为止，似乎没有什么思路，user页可控，不过连标签都插不了，这时候可以思考一下之前发现的奇怪的地方，.././classes.css

随手试了下路径发现了一个问题，http://52.80.19.55/classes.php/1/2/3/ 加载了 http://52.80.19.55/classes.php/1/classes.css 而/classes.php/1/classes.css显示的是/classes.php/1/的内容，这点就非常有趣了，我们可以通过多级路径使得classes.css变成一个php，那么有没有可能让他指向我们可控的user.php，通过css标签比如引入一个background-image，这样就能实现bot访问我们服务器的目的。（嗯，就是这个点我打3分，我一直以为这样是不可以的，因为css不像html，按照规范，必须严格解析不允许有错误的，chrome的包容性这么强是我没想到的）

接下来就要解决控制classes.css的问题，我们需要让整个网页显示classes.php，而classes.css指向user.php，这里我想到了urlencode，具体解释在上面，我是在这时候才想起用 %2F， %2E 去测试，发现浏览器不认 %2E，会强制转为 .，而 %2F 成功的骗过了浏览器和nginx，讲道理chrome和nginx一样需要背锅，因为文件名本来就不允许包含 /，为什么不强制把 %2F 转换为 /

接下来要解决css解析的问题，毕竟我们引入的css是一个网页，这个可以在本地直接测试，当然直接用 `{background-image:url (xxx);}` 是不会生效的，我测试了 `*/*{background-image:url (xxx);}`，`-->*/{background-image:url (xxx);}`，(构造一个标签尾，让chrome去关闭忽略前面的，不过没有效果) `*/{background-image:url (xxx);}*/{background-image:url (xxx);}` `{background-image:url (xxx);}`（这个成功了，chrome竟然识别了）

构造payload

首先构造url，这个知道原理就很容易构造，随手构造了一个 `http://52.80.19.55/user.php/271/2/%2f..%2f..%2f..%2fclasses.php%2f1%2f/`，这样classes.css就会变为/user.php/271（ID要改）

注册一个账号，构造vow，我精简了下，用了 `*/*{background-image:url (http://vpsip);}*/{}`

然后提交构造的url

□

出题

在这之前，先说我自己出题的一件事，我第一次出大型赛事的CTF题，也就是BCTF2016，这年我出的题被Riatre大佬吐槽了，还被众多老外吐槽了，基本上被钦定为guessing题了，事后被各种教育应该怎么出题，当时众大佬是这么教育我的，CTF是一个游戏，应该让做题人感觉到玩得开心，经过自己的努力，能够学到东西，能够**收获做出题的喜悦**，而不是白白浪费时间，还什么都没学到。

嗯，说的很有道理，道理我都懂，不过怎么做。其实这也是这1年来我思考的问题，怎么出好题。

我自己总结了三条，不考弱智知识点，不设置不必要的障碍，**给做题人方向上的引导**，三点可以在归结为一条，站在做题人的角度思考问题。

- 不考弱智知识点：什么是弱智知识点，就是你做出来觉得毫无收获的东西，比如爆破密码，站在做题人的角度，你一点也不希望遇到这些毫无卵用的考点吧
- 不设置不必要的障碍：出题者的大忌，“我要让做题者做不出题”，比如让做题人猜后台路径（对，即使是fuzz后台也要尽量避免，这个度怎么掌握呢，就是你自己来做，你手试，你能试出来，比如robots.txt，比如.git），猜加密密码，或者密码过长，导致hash只有cmd能解密，没法本地跑出（这我犯过的错误），站在做题人的角度，你遇到这样的题，你会不会掏刀捅出题人。
- 给做题人方向上的引导：用人话说，你得让做题人知道他该做什么，而不是把时间浪费在无用的测试上，比如一道注入题，把题出的非常像xss题，导致做题人花了大量的时间测试xss，最后才发现是注入，那做题人浪费的那些时间是不是白白浪费了，是不是什么都没学到，何来

喜悦？所以说这一点，也是我觉得最重要的一点，你必须要让做题人知道，他要干什么，他可以在这个方向去花他的时间，想办法，怎么绕过，怎么去实现xxxxxx，这样他的时间没有被浪费，最终查阅资料，学到了姿势，做出题了，是不是很爽。所以对出题人而言，必须想办法暗示做题人，你该做啥，甚至直接在题目里说明。

以本题为例，犯的错误就是第三点，作为出题人，你一定要假想你是做题人，而且你是不知道出题人要你做啥的，所以你拿到题，会一遍分析，发现这题明显是xss题啊，绕过过滤吧，嗯我就是这样的，我花了很多时间想怎么xss成功，甚至在有烦人的验证码的情况下去盲测那个comment能不能xss，还要查资料，是不是有最新的黑科技可以干成？结果呢，我做出来后是不是想打人。所以对这题而言，如果能够委婉的提示bot的ua含有flag，就能让做题人少浪费很多时间。

而在我接下来的出题中，基本开始贯彻这个思想，比如AliCTF的homework，我在readme.html里暗示了用注入来替换缓存是能够实现的，Resume System里可以用nmap发现11211端口是filtered的，提示用ssrf攻击memcached，比如还是BCTF被吐槽的那题，如果在题目主干加上pentest，同时不设置那么变态的hash，就不至于被喷的那么惨。而pwnhub我出的那道题就又犯了错误，应该把hint（admin只是普通用户）直接放在题目主干上，或者改admin为firesun。这样可以加快做题人发现是self-xss的进度，把时间投入到有意义的事上。

当然其实出题还有很多要注意的比如难度控制，考点数量控制等等，这里就不细说了，还是那句话，多站在做题人的角度思考问题。

今年BCTF，准备了1-2题，个人觉得难度中偏难，还请各位大佬到时候多提意见。也希望众出题人能够在将来多出一些高质量的题目，Orz



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)