web_ctf 题型总结

原创

昂首下楼梯 于 2019-09-28 18:07:45 发布 3211 收藏 83

文章标签： ctf 入门级 web

本文链接：https://blog.csdn.net/qq_42812036/article/details/101604671

ctf 常见套路总结

01、 工具集

基础工具
-Brupsuite
-Python
-Fiorefox

扫描工具
-御剑
-Nmap
-AWVS

文件上传工具
-菜刀、cknife

文件包含工具
-LFIsuite

暴力破解工具
-bp暴力破击
-Hydra

其它一些比如、010 Editor、wireshark

02、 常用套路总结

- 查看html代码
  火狐的F12，利用好，能看到的可能直接就是flag，也可能一道ctf题在页面中没有任何解题有关提示，但可能在源码中注释部分会有提示。

- 查看http数据包
  当然不止有源码注释可以隐藏信息,用bp抓包，查看http请求/响应，也是很重要的一个方法

- 修改或添加HTTP请求头
  常见要修改的有
  -Referer: 来源伪造
  -X-forwarded-For：ip伪造
  -usage-agent:用户代理
  -cookie：身份识别伪造

这些修改都很基础，主要的是要能识别出来，要要我们伪造或这修改的是什么。
就比如改referer,做题之外也要想想怎么来预防来源伪造，这是做题的深层含义。

-

web源码泄露

- https://www.secpulse.com/archives/55286.html
- 下面是4中常见的

-Vim源码泄露

> 解题思路：如果发现页面有提示vi或者vim，说明存在swp文件泄露 那么伪造地地址：/.index.php.swp或者/index.php-

如果源码down下来乱码，可以使用Linux：

```
vim -r index.php
```

-备份文件泄露
地址:
index.php.bak
www.zip
wwwroot.zip
htdocs.zip
.rar
.zip
.7z
.tar.gz
.bak
.swp
.txt

-.git源码泄露
地址： `http://www.xxx.com/git/config`

工具：GitHack、dvcs-ripper

-SVN源码泄露

> SVN（subversion）是源代码版本管理软件。在使用SVN管理本地代码过程中，会自动生成一个名为.svn的隐藏文件夹，其中包含重要的源代码信息。但一些网站管理员在发布代码时，不愿意使用‘导出’功能，而是直接复制代码文件夹到WEB服务器上，这就使.svn隐藏文件夹被暴露于外网环境，黑客可以借助其中包含的用于版本信息追踪的‘entries’文件，逐步摸清站点结构。

地址： `http://www.xxxxx.com/svn/entries`

工具：dvcs-ripper、Seay-Svn

-.hg源码泄露
漏洞成因：hg init的时候会生成 .hg
工具 dvcs-ripper
也可以在网站根目录下搜索 .hg

-.DS_Store源码泄露

- **编码加密解密**

> 编码类的题主要还是看经验，在这种算下面的几种算很基础的。

-Base64多次加密
Py解密脚本

```
import base64
fp=open('1.txt','r')
a=fp.read()
while 1:
 a=base64.b64.decode(a)
 print a
```

-摩尔斯电码
它的组成有：
点、划、空格、斜杠/

-培根密码
实际是用二进制来的
通常用a和b来表示，选特定个数为一组，对应明文的一个字符

-栅栏密码

-凯撒密码
用偏移量来代表凯撒密码把，比如偏移量为3
密文中的a就对应明文的d,b就对应e

-JsFuck

- Windows特性
  -短文件名
  利用“~”字符猜解暴露短文件/文件夹名。
  例如
  例如：backup-082119f75623ev4df7fds6545ff66fsd.sql，其短文件名是 BACKUP~1.sql

---

-IIS解析漏洞
绕过文件上传检测
服务端白名，黑名单

下面是进阶一点的

003、 php弱类型

1. **PHP包含的类型**
   -string
   -integer
   -array
   -double
   -boolean
   -object
   -resource
   -NULL

==遇到不符合类型，自动转换

2. 知道这些可以来了解下PHP类型比较，
   在==情况的比较下，在比较前会将字符串类型转换成相同的再比较。如果涉及到比较数字和字符串，会转换成数字再比较。===不进行转换，先判断类型。
   常见的例子：

> ‘123’ = 123
> ‘0x01 == 1 #十进制数转换成数字1
> '' == 0 == false
> NUll == false == 0
> [false] == [0] == [null] ==[”]
> true == 1
> ‘abc’ == 0
> ‘123a’ == 123
> ‘0e123456789’ == ‘0e987654321’

一些见过的题：

> 题型1 strcmp字符串比较

```
define('FLAG','pwnhub(THIS_IS_FLAG)');
if (strcmp($_GET['flag'],FLAG == 0)){
 echo "success, flag:" . FLAG;
}
```

> -strcmp(string $str1, string $str2)
> -如果str1<str2,返回<0;如果str1>str2,返回>0;如果两者相等，返回0
> -flag[]= xxx =>strcmp比较出错==>返回NULL => **NULL == 0** => Get FLag!

> 题型2 MD5绕过

```
define('FLAG','pwnhub(THIS_IS_FLAG)');
if ($_get['s1'] != $_get['s2'] && md5($_get['s1']) == md5($_get['s1'])) {
 echo "success,flag:" . FLAG
}
```

> 其原理就是：原值满足不相等，但md5加密后值相等，**这就是弱类型**

有两种解法，第一种用科学计数法，第二种用数组trick

ctf中常见的PHP弱类型总结请见
https://www.cnblogs.com/anbus/p/10000571.html

004、 绕 WAF

1. 绕过WAF的几种常见方法#

- 大小写混合
  形式：`uNion sEleCt 1,2,3,4,5 fRoM admin`
  -用于只针对大写或小写的关键字匹配技术，正则表达式匹配是大小写不明干（/i）无法熬过
- 使用编码（url编码、16进制）
  ’ 为%27
  /为%2f

```
union sleect 1,table_name from information_schema.tables where table_schema=**数据库名字(16进制)**
```

- 使用注释
  常见的注释符号

> //，–，--+，#，-- -，;–a
> /**/在构造的查询语句中插入注释对规避空格的依赖或关键字识别
> //、#、–等用于注释后面的语句

> eg:
> //**union**//select//**1,2,3,4,5**//from//**admin**
> **也可以试下这个**
> //un//**ion**//sel//**e**//ct//**1,2,3,4,5**//from/**/admin/*!and*/ 1=2

- 使用空字节
  id=1 %00 and 1=2
- 使用嵌套剥离
  selselect 剥离后 select
- 避开自定义过滤器
  例： and 转换为 a+nd、a%nd 、‘a’nd 、%A0and

附一张图