参加运维赛和领航杯的自闭历程

原创

m0_45118974 于 2019-11-24 21:47:31 发布 261 收藏 1

周三周四是高校运维赛，周三下午的理论题靠自己吧安全知识不够，靠度娘吧搜不到……佛了。周四一整天的ctf，

首先是Misc3，给了一串&zwnj和&#8203，就想着怎么给它解码吧，说是html编码，python脚本解码，解出来啥也没有，百度发现这俩东西一个是&zwnj（零宽不连字） 说明：&zwnj是一个不打印字符，放在电子文本的两个字符之间，抑制本来会发生的连字，而是以这两个字符原本的字形来绘制。另一个&#8023是零宽度空间字符，没有可见的字形，没有宽度，不会再视觉中发现任何内容。佛了……搞了这么久告诉咱这是不可见的东西。仔细看这一串东西是交替出现并且只有它俩，所以应该是把&zwnj用0表示，&#8203用1表示，八个一组二进制转ascii得到flag！

其他题目就参考大佬的writeup记录一下吧……

ezbypass

打开看到源码，这么简单的源码没想到这么变态（好吧我承认是我技术差）。

大佬的writeup……明人不说暗话我看不懂啊，讲真这么长个php文件我啥时候也能自己写啊

大佬牛B

查看下phpinfo(),ban了error_log,mail，以及putenv之类的，常规的bypass disable_functions应该是莫得了，bypass 下open_basedir发现/readflag，应该是要rce，想起前段时间看的硬核bypass

https://github.com/mm0r1/exploits/blob/master/php7-gc-bypass/exploit.php

tmp目录下写个shell 文件包含连上

在tmp目录下自己再写个php文件

1.

2. <?php

3. pwn("/readflag");

4.function pwn($cmd) {

5. global $abc, KaTeX parse error: Expected 'EOF', got '&' at position 32: …nction str2ptr(&_str, $p = 0, $s = 8) {

7.

$address= 0;8. for(j = $s-1; $j >= 0; $j–) {

<?php #error_reporting(0); session_start(); include "config.php"; $username = $_POST['username']; $password = $_POST['password']; if (isset($username)){ $sql = "select password from user where name=?"; if ($stmt = $mysqli->prepare($sql)) { $stmt->bind_param("s", $username); $stmt->execute(); $stmt->bind_result($dpasswd); $stmt->fetch(); if ($dpasswd === $password){ $_SESSION['login'] = 1; header("Location: /upload.php"); }else{ die("login failed"); } $stmt->close(); } }else{ header("Location: /index.php"); } $mysqli->close(); 就很懵知道吧因为看不懂 以下为大佬的做法 查看源码，发现经过了预处理（预处理是啥？我去康康），但是发现$dpasswd没有定义，为NULL，$_POST['password']不对其传参，也为NULL，于是构造如下请求，绕过身份验证。 ![在这里插入图片描述](https://img-blog.csdnimg.cn/20191124162627506.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L20wXzQ1MTE4OTc0,size_16,color_FFFFFF,t_70)重定向到了上传页面，经过测试发现，上传点校验了文件头和文件后缀名，后缀名改为php7，文件头写为GIF89a ![在这里插入图片描述](https://img-blog.csdnimg.cn/20191124162812919.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L20wXzQ1MTE4OTc0,size_16,color_FFFFFF,t_70)服务端没有对上传文件重命名，添加Shell，http://111.186.57.61:10501/uploads/1337.php7成功连接。 ![在这里插入图片描述](https://img-blog.csdnimg.cn/20191124162835695.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L20wXzQ1MTE4OTc0,size_16,color_FFFFFF,t_70)成功的到flag ![在这里插入图片描述](https://img-

blog.csdnimg.cn/20191124162929192.png) 现在我想的是大佬牛B，顺便哪个大佬把环境复现一下我跟着搞一遍啊…… 说起领航杯也是辛酸泪啊 先说lsb这题吧，图片隐写题，下载工具stegsolve，先补习一下lsb知识：
https://blog.csdn.net/qq_33438733/article/details/79324763 LSB即最低有效位。LSB加密是信息隐藏中最基本的方法，由于人们识别声音或图片的能力有限，因此我们稍微改动信息的某一位是不会影响我们识别声音或图片的。 通常来说LSB加密用在无损压缩的数据格式文件中，例如图像中的bmp格式和音频的wav格式。由于这两种格式未对源数据进行有损压缩，因此可以将信息隐藏起来。 lsb在BMP文件中的使用： 对于图像文件LSB的特征很明显，通常将信息隐藏在某一个颜色通道中。我们可以查看图片的每个像素点的RGB值，或者使用stegsolve工具进行查看。 由于图像是有像素构成的，每个像素有8位（对于BMP图像来说），通常最后一位的变化，通过肉眼是无法察觉的。 这位大佬讲的也很棒：
https://blog.csdn.net/qq_37414405/article/details/84651714 但其实这道题的做法是：把图片在stegsolve中打开，一直按左箭头，找到flag 给我深刻印象的还有明文破解题 首先根据题目描述暴力破解了level1的密码，发现level1中有Readme.txt和level2.zip，发现level2.zip 里有一个加密的level3.zip和Readme.txt，那么就开启明文破解模式。将level1.zip里的Readme.txt压缩成Readme.zip,用arpach工具进行明文破解level2.zip;虽然我的电脑没跑出来，但讲真我觉得我做的没错；学弟电脑跑出来了level3.zip里有一个加密的flag.png，改了长宽后得到flag。 ![在这里插入图片描述](https://img-blog.csdnimg.cn/20191124214653979.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L20wXzQ1MTE4OTc0,size_16,color_FFFFFF,t_70)搞了一个该长宽的例子。