CTF技巧_Web——PHP特性_绕过ereg()

原创

Ho1aAs 于 2020-11-21 21:15:26 发布 501 收藏

分类专栏： # Tricks 文章标签： php 正则表达式 安全 ctf web

文章目录

一、简述

`int ereg(string pattern, string originalstring, [array regs])`函数用指定的模式搜索一个字符串中指定的字符串,如果匹配成功返回t1,否则,则返回0，且搜索对大小写敏感

`ereg()`函数存在NULL截断漏洞，当传入的字符串包含%00时，只有%00前的字符串会传入函数并执行，而后半部分不会传入函数判断。因此可以使用%00截断，连接非法字符串，从而绕过函数

二、PHP源码

> 题目来源：ctf.show——web入门_web108

```php
<?php

/*
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date:   2020-09-16 11:25:09
# @Last Modified by:   h1xa
# @Last Modified time: 2020-09-28 23:53:55

*/


highlight_file(__FILE__);
error_reporting(0);
include("flag.php");

if (ereg ("^[a-zA-Z]+$", $_GET['c'])===FALSE)  {
    die('error');

}
//只有36d的人才能看到flag
if(intval(strrev($_GET['c']))==0x36d){
    echo $flag;
}

?>
```

三、方法及原理

首先需要GET参数，参数的值要是`0x36d`，包含字母，且参数要反向。通过计算：0x36d=(877)D，反向后是778，最后使用%00截断

即：`payload:?c=a%00778`

执行GET请求时会默认进行一次URL解码，由于%00是一个不可见字符，c的值实际是`a(invisible)778`。经过`ereg()`截断，前部分`a`通过匹配返回1，不会`die('error')`。再反向取值，0x36d比较后部分的`877`返回1，从而`echo $flag`