# **green CTF 2018-1**

**合作文档**

http://124.16.75.162:40009

[关于本文档](#_hhgek7d344f4)

[如何使用本文档](#_k9cje1mijw1h)

[XXX | OPEN | working :](#_8o52cmn5kevk) lwj

[基本信息](#_seex8uohl039)

[官网地址](#_37hfbsg9znsx)

[比赛账号](#_x3uksrey37dp)

[起止时间](#_9ky8h2e60wx2)

[题目](#_8tj3h3320ax)

# **关于本文档**

使用本文档来促进线上的队员和在比赛基地的队员之间的合作，并记录整个比赛的过程。

## **如何使用本文档**

* 题目状态：
  1. OPEN - 正在试图解这道题
  2. CLOSED - 这道题还没有打开
  3. SOLVED - 解决了！鼓掌撒花！
* 根据个人能力，可以选择做还没有解出的题来挑战自我，即使这题已经被解出，依然可以继续做这题来提高自己（不过强力建议先做未解得题，因为事后会有wp整理），不过不管怎样，希望大家做到以下几点：
* **不要只顾一人做题，不看文档，不写文档**
* 解一道题，不管这题是否解出，请把**你的名字加入到Working列表**
* 如果你卡住了，或者解出这题，请先把writeup或者目前的进展写到文档里再做下一题，如果是解出了，**请在写完wp后贴出flag，并且更新题目状态，更新之后请刷新页面最上方的目录**，书写格式如下（例子）：**（比赛中可以写的很简洁，可以就写思路或主要步骤即可，但是不要不写）**

**题目格式：**

**====================================**

### **XXX | OPEN | working : zy**

**===================================**

* 如果你是做已经解决的题目，建议先自己尝试，直到做不出来才看看别人的思路，如果你有别的方法，同样把wp写在文档里，格式与上相同
* **比赛需要的是团队合作，请看重文档工作，从第一次合作开始就遵守规则，做完一题就做一个总结，这对大家都有帮助**
* 比赛结束，如果有些题在比赛中只写的简单的wp，可以花时间补充完善，写wp其实对个人的帮助一样很大
* 比赛期间欢迎讨论，目前先使用qq群交流，也可以直接在doc上交流
* WP比赛后会整理给大家学习，希望大家都能在比赛后在认真总结一番

# **基本信息**

## **官网地址**

http://124.16.75.162:40009

## **比赛账号**

Username: green

Password: green20170113

## **起止时间**

中国标准时间（China Standard Time） **1/13 9:00 ~ 1/13 20:00 共11小时**

# **题目**

web题目

## babySqli| open | working:wonderkun

被禁止的字符： %00 , # , ; / , \* , & , | , -- , @ , 空格 , %09 , %a0, %0b,%0c,%0d,%a0, ` , , () , ' , " , ^ , %

Mmp,不写了，意思就是你能想到的字符，他全都想到了。。。。。

Hint.php 貌似可以用 %0a

## babyXnuca | solved | working : wonderkun

存在xxe漏洞 ，读取文件

#index.php

<?php

//ini\_set("display\_errors", "On");

//error\_reporting(E\_ALL | E\_STRICT);

error\_reporting(0);

if(isset($\_POST['data']) and $\_POST['data'] != "") {

$xml = simplexml\_load\_string($\_POST['data'], null, LIBXML\_NOENT);

}

///tmp/index.php.bak

?>

读取了 /tmp/index.php.bak

<?php

error\_reporting(0);

$file=$\_GET['file'];

if(strstr($file,"../")||stristr($file, "tp")||stristr($file,"input")||stristr($file,"data"))

{

echo "Oh no!";

exit();

}

include($file);

?>

有内网内网的代码是/tmp/index.php.bak

Php的任意文件包含做任意读取，获取到俩文件

Log\_hahahaha.php

<?php

// error\_reporting(0);

class Awd

{

public function \_\_construct($filepath)

{

$this->filepath = $filepath;

}

public function Flow()

{

$HTTP\_Method = $\_SERVER['REQUEST\_METHOD'];

if(!file\_exists($this->filepath))

{

mkdir($this->filepath,0777);

}

$filename = $this->filepath."access.log";

$request\_url = $\_SERVER['REQUEST\_URI'];

$protocol = $\_SERVER['SERVER\_PROTOCOL'];

$ip = $\_SERVER['REMOTE\_ADDR'];

$time = date('Y/m/d h:i:s');

$content = $ip."\t".$time."\t\n".$HTTP\_Method.' '.$request\_url.' '.$protocol."\r\n";

if (isset($\_GET['log'])) $content=urldecode($\_GET['log']);

file\_put\_contents($filename,$content,FILE\_APPEND);

echo $content;

}

}

if(isset($\_GET['auth'])){

if(preg\_match("/[^A-Za-z]/",$\_GET['auth']))

{

die();

}

}

else{

die();

}

if(file\_exists("./log/".$\_GET['auth']."/access.log"))

@unlink("./log/".$\_GET['auth']."/access.log");

$Catchs = new Awd("./log/".$\_GET['auth']."/");

$Catchs->Flow();

?>

dump\_lalalalala.sql

DROP TABLE IF EXISTS `flag`;

CREATE TABLE `flag` (

`flag` VARCHAR(100)

);

CREATE USER 'rebirth'@'localhost';

GRANT USAGE ON \*.\* TO 'rebirth'@'localhost';

GRANT SELECT ON `flag`.\* TO 'rebirth'@'localhost';

写个shell，读取数据库，就拿到flag了

## babyAuth | solved | working : AFang,wonderkun

https://auth0.com/blog/critical-vulnerabilities-in-json-web-token-libraries/

伪造token过校验。

# RE

## HexIt| solved | working:lmx、giglf

Checksum算法

十六进制文件界限不清晰

写了个脚本修复每一行的checksum，然而并没有什么发现

这是个Arduino单片机的东西，并不懂

出题人说：把ida用对就能准确反汇编，这个题反汇编出来不需要动态调就拿到flag了

Arduino **avr** 指令集，莫名其妙翻到了原题，flare-on2017的原题改了数据，挺有意思的**Flare-On**

<https://www.fireeye.com/content/dam/fireeye-www/global/en/blog/threat-research/Flare-On%202017/Challenge9.pdf>

官方wp给了静态、模拟器动态、真机三种解法Orz

## babyre| open | working:giglf

发现MD5的初始化值，对输入hash后再对比，但我电脑缺少dll无法正常运行

无法调试

## easyre| solved | working:giglf、lmx

Sub\_401750 return true->correct! But not flag

关键函数sub\_6DE530 后面直接花，接33333个分支，不过每个分支的操作都是一样的

跟到0x0047382E有惊喜，还在分析

后面异或了一段汇编码，进入了正确的判断逻辑，最后会修改上面用于判断是否正确的那个变量

wp赛后补

PWN题目

## chaniuka| open | working:yjw

update\_info泄露出libc信息

## dictor | solved | working: AFang

格式化串，就一个点，没有栈溢出.

但是是&buf，所以不能直接修改main函数返回地址，

可控一些指向libc的变量。

Printf会在某些情况下调malloc.

先劫持执行流，跳转到start，构成一个循环，

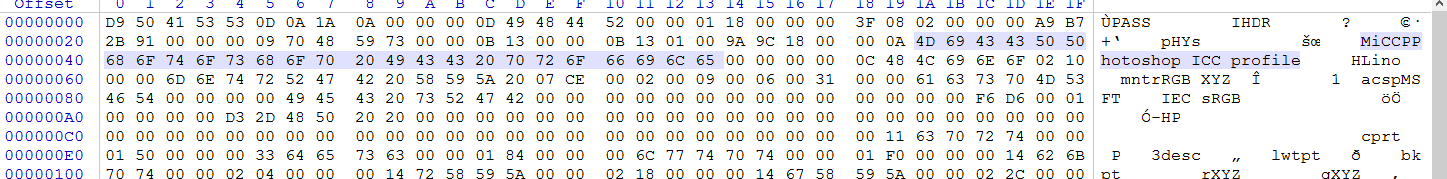
这个时候没办法，必须靠malloc去再度劫持执行流。

## mahoshijio| solved | working:yjw

Uaf + double free

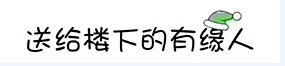
MISC题目| open | working: Nevv,LMX,AFang

Binwalk 查看jpg后边zlib compressed data，解压后感觉像是残缺的IDAT块



FF D9是文件结尾

后边是另一张图片,你改下头四个字节为 89 50 4E 47



隐写的工具用jphs吧，密码应该就在这张图片里

有人接着接续不，我要回学校啦。。。