NCTF-Writeup

原创

Yukikaze_cxy 于 2018-05-30 22:41:26 发布 1315 收藏

南京邮电大学网络攻防训练平台

https://cgctf.nuptsast.com

以下按本人做题的顺序排序

剩余题目待补完。。。

【Web】
1.签到题
nctf{flag_admiaanaaaaaaaaaaa}
网页源码

2.单身二十年
nctf{yougotit_script_now}
由于页面自动跳转，使用工具查看search_key.php页面源码即可

3.SQL注入1
nctf{ni_ye_hui_sql?}
考虑到pass使用md5加密，故在user上注入。由于使用trim去除空格，而mysql除了--外还可使用#作为注释，参考source中SQL写法注入即可

4.SQL注入2
ntcf{union_select_is_wtf}
pass仍然使用md5加密，仍在user上注入。考虑到题目提示的union，联想到令user为空，union select 一段md5值（如123456的md5值），再在pass栏输入原值（123456）即可

5.签到题2
nctf{follow_me_to_exploit}
输入框限制10单位长度，F12审查元素将其改为15后，输入提示即可

6.这题不是WEB
nctf{photo_can_also_hid3_msg}
拖下gif，使用十六进制查看器在文件末尾找到

7.php decode
nctf{gzip_base64_hhhhhh}
将php文件中eval改成echo，执行即可

8.AAencode

nctf{javascript_aaencode}
由于页面读取乱码，将txt文件下载下来，分离出最后一句。先执行前面的，再执行最后一句，放入浏览器console执行即可

9.COOKIE
nctf{cookie_is_different_from_session}
使用fiddler查看发送的数据报文头部，复制内容并将cookie中的login的值0改为1，提交观察返回即可

10.单身一百年也没用
nctf{this_is_302_redirect}
使用fiddler查看发送的数据报文头部即可

11.md5 collision
nctf{md5_collision_is_easy}
借助php处理结果为0e开头的hash函数会误认为其为科学计数法的bug
http://www.freebuf.com/news/67007.html
另：参考王小云博士关于哈希碰撞的研究报告

12.层层递进
nctf{this_is_a_fl4g}
F12打开开发者工具，全部展开，找到一个404.html，双击被注释掉的内容使其格式化后发现flag

13.GBK Injection
nctf{gbk_3sqli}
根据提示，字符集为GBK，考虑在sql结尾加上%d5以将自动补全的\转义掉，同时使用%23(#)注释掉后面的内容（直接用#也会被转义）。默认查询id为1，改为2,3后得到提示the fourth table。参考4，使用union select 1,2，发现返回的是第二个字段。构造sql：union select 1,table_name from information_schema.tables limit n,1查看表名，发现limit 43,1为ctf4表，count该表发现仅1行。由于单引号会被转义，考虑使用ctf4的十六进制值0x63746634，构造sql：union select 1,column_name from information_schema.columns where table_name=0x63746634 limit n,1爆出全部列名，发现第二列为flag列。
考虑到该表仅1行，于是有sql：select flag from ctf4。完整构造语句见下
id=%d5%27%20union%20select%201,flag%20from%20ctf4%20%23where%20table_name=0x63746634%20li

14.bypass again
nctf{php_is_so_cool}
php中的弱类型问题，==为类型转换后若相等则为true，===必须完全相等。参考11，处理0e开头的字符时，php会认为其为科学计数法，故构造2个字符md5值为0e开头即可

15.综合题
nctf{bash_history_means_what}
复制编码内容放在console中执行，得到1bc29b36f623ba82aaf6724fd3b16718.php，打开页面获得提示“在脑袋里”，在response的header中找到tip：bash history。查看当前目录下的.bash_history文件得到zip -r flagbak.zip，下载该文件并解压得到flag
编码方式参考http://www.freebuf.com/sectool/5352.html

16./x00
nctf{use_00_to_jieduan}
ereg函数为正则匹配，但是根据提示使用/x00即%00可将其截断（ereg漏洞）。又要求传入数据包含#biubiubiu，由于#会将后文内容全部注释，故考虑采用%23代替#，即构造传入参数1%00%23biubiubiu

17.变量覆盖【2】

17.变量覆盖【？】
nctf{AD3FBD8D5928693CA499347C91570AE6}
使用fiddler查看返回的数据报文，指定传入name=meizijiu233即可

18.变量覆盖
nctf{bian_liang_fu_gai}
查看源码，发现post方法有两个参数，pass和thepassword_123。传递参数时指定两者为相同值即可在页面上获得flag

19.起名字真难
nctf{follow_your_dream}
根据源码，要求输入不可为1-9的任意数字，且结果等于54975581388。考虑十六进制转化，发现该值刚好为0xccccccccc，传递该值即可

20.SQL Injection
nctf{sql_injection_is_interesting}
考虑MySQL中\'可转义单引号，根据源码分析发现只要有结果返回即可得到flag。指定username参数为\以转义掉后面的单引号，考虑到password参数的第一个引号将用来闭合username未闭合的引号，同时也为了返回结果，故指定password参数为or 1 #以略去后面的引号，此时又发现#也被过滤，改用%23可获得结果

【Crypto】
1.easy
nctf{this_is_base64_encode}
base64加密，解密即可

2.Keyboard
nctf{areuhack}
键盘字母排列

【Misc】
1.图种
nctf{dssdcmlw}
根据提示，修改后缀名，解压缩得到新的gif

2.Remove Boyfriend
flag{who_am_1}
解析采集的包，得到flag.py，执行发现得到{flag_is_not_here}，同时拼接出一张图，将图中的synt{jub_nz_1}替换flag.py中的{synt_vf_abg_urer}，得到{flag_who_am_1}，改写一下即可

3.MD5
nctf{e9032994dabac08080091151380478a2}
【投机取巧】暴力破解法
考虑：MD5加密算法手动计算