INS'hAck CTF 2018 - base65535

Apr 8, 2018 • jbz • Crypto

> *Base 64 is too mainstream but we could not manage to decide on the final encoding scheme… https://base65536.ctf.insecurity-insa.fr*

There is a server at `base65536.ctf.insecurity-insa.fr` that answers `Method Not Allowed`.
By doing a POST request we obtain `Param 'sample' is expected"`

Let’s try to send this parameter:

```
$ curl -X POST 'https://base65536.ctf.insecurity-insa.fr/' -d "sample=aabbcc"
Your encrypted sample is □緩̈ and the flag is ᒥ-□捏棂흰ː̇ń鶩輫□ƨ !
```

This program writes the sample and the flag in hexadecimal:

```
#include <stdio.h>
#include <string.h>

int main(){
    int c,i;
    unsigned char buf[100000];
    for(i=0;(c=getchar())!=-1;i++){
        buf[i]=c;
    }
    int end2=i;
    unsigned char *my = buf+25;
    unsigned char *myend = memmem(buf,100000," and the",8);
    int eq=0;
    int len = myend-my;
    unsigned char *flag = my+len+17;
    int len2 = end2-(flag-buf);
    for(i=0;i<len;i++)printf("%02x ",my[i]);
    printf("\n");
    for(i=0;i<len2;i++)printf("%02x ",flag[i]);
    printf("\n");
}
```

```
$ curl -X POST 'https://base65536.ctf.insecurity-insa.fr/' -d "sample=aabbcc" |
e4 81 be e0 aa aa e1 8a b4
eb a4 bc e1 ab a6 f0 95 94 95 f0 95 a5 b7 f0 94 a1 83 e5 bf 88 f0 95 87 9d eb 96
```

By trying some inputs, we can see that some patterns repeat: `aaaaaaaa`

`aa|aa|aa|aa: f0 93 bd 86 | f0 93 bd 86 | f0 93 bd 86 | f0 93 bd 86`

`aabbaabb`

`aabb|aabb: ed 83 98 eb 8d 82 | ed 83 98 eb 8d 82`

`aabbccaabbc`

`aabbcc|aabbcc: e1 9e a1 eb 97 9d c8 b7 | e1 9e a1 eb 97 9d c8 b7`

We can notice that the same pair of character corresponds to the same output. So we can precompute a table with all possible encrypted values by performing a request for all pairs of letters and then use the precomputed values to decrypt the flag.

By sending a request containing the sample `...AA112211AB112211AC112211AD112211AE112....zz112211` we obtain something containing a repeating pattern (`e7 b3 b4 e1 a1 80 e7 b3 b4` in our case) that correspond to `112211` so we can split the string on it and recover the encrypted value for each pair from `AA` to `zz`.

```
$ curl -X POST 'https://base65536.ctf.insecurity-insa.fr/' -d "sample=AA112211AB1
$ cat out | ./aaa | sed 's/e7 b3 b4 e1 a1 80 e7 b3 b4/\n/g' > output1
$ paste list_of_pairs output1 > solution_list
...
AA  f0 93 81 9a
AB  e5 93 b7
AC  e9 bd bf
AD  ec 99 9b
AE  f0 93 97 a9
AF  f0 96 b3 b6
AG  f0 96 a9 ae
AH  e2 85 97
AI  ef 94 b1
AJ  e9 9d 84
...
```

The flag sent by the server is:

```
e8 89 bd e1 96 bf f0 91 b5 8d ee 9e bf e4 88 9c f0 91 b7 a2 e1 80 9f ea a2 bc f0
```

Now we can decrypt it by searching in the solution list, for example we can search `e8 89 bd` and notice that the flag starts with `IN`.

```
$ cat solution_list | grep "e8 89 bd"
IN  e8 89 bd
```

Related Posts

« Prev

Next »