The test in Linux we put at the end of the report

Võ Anh Kiệt – 20520605

Nguyễn Bảo Phương – 20520704

Phase 1:

The pass is easy to get is

The moon unit will be divided into two divisions.

```
int __cdecl phase_1(int a1)
{
  int result; // eax@1

  result = strings_not_equal(a1, "The moon unit will be divided into two divisions.");
  if ( result )
    explode_bomb();
  return result;
}
```

Phase 2:

This is the function in the assembly

```
  v4 = *MK_FP(__GS__, 20);
  read_six_numbers(a1, v3);
  if ( v3[0] < 0 )
    explode_bomb();
  for ( i = 1; i <= 5; ++i )
  {
    if ( v3[i] != v3[i - 1] + i )
      explode_bomb();
  }
```

We will re-write the code, however the code with the condition >= 0 with not get the bomb so we decide to run from 0 to 10

```cpp
// Example program
#include <iostream>
#include <string>
#include <stdio.h>

int main()
{
    int v3[6];
    for (int j = 0; j <= 10; ++j)
    {
        v3[0] = j;
        std::cout << v3[0] << " ";
        for (int i = 1; i <= 5; ++i)
        {
            v3[i] = v3[i - 1] + i;
            std::cout << v3[i] << " ";
        }
        std::cout << "\n";
    }
}
```

```
0 1 3 6 10 15
1 2 4 7 11 16
2 3 5 8 12 17
3 4 6 9 13 18
4 5 7 10 14 19
5 6 8 11 15 20
6 7 9 12 16 21
7 8 10 13 17 22
8 9 11 14 18 23
9 10 12 15 19 24
10 11 13 16 20 25
PS D:\test>
```

Get the first list is password

0 1 3 6 10 15

Phase 3:

```
v6 = __isoc99_sscanf(a1, "%d %c %d", &v4, &v2, &v5);
if ( v6 <= 2 )
  explode_bomb();
switch ( v4 )
{
  case 0:
    v3 = 102;
    if ( v5 != 995 )
      explode_bomb();
    return result;
  case 1:
    v3 = 106;
    if ( v5 != 726 )
      explode_bomb();
    return result;
  case 2:
    v3 = 115;
    if ( v5 != 694 )
      explode_bomb();
    return result;
  case 3:
    v3 = 101;
    if ( v5 != 515 )
      explode_bomb();
    return result;
  case 4:
    v3 = 111;
    if ( v5 != 846 )
      explode_bomb();
    return result;
  case 5:
    v3 = 112;
    if ( v5 != 521 )
```

With this function it is easy to get the list that

First number is 0 to 7

Second char is the letter converting from ascii number

Last number is shown in the code

The password is 7 of this list:

0 f 995

1 j 726

2 s 694

3 e 515

4 o 846

5 p 521

6 v 784

7 b 778

Phase 4:

We re use the func4

```
int func4(int a1, int a2)
{
    int result; // eax@2
    int v3;     // ebx@5

    if (a1 > 0)
    {
        if (a1 == 1)
        {
            result = a2;
        }
        else
        {
            v3 = func4(a1 - 1, a2) + a2;
            result = v3 + func4(a1 - 2, a2);
        }
    }
    else
    {
        result = 0;
    }
    return result;
}
```

Then we rewrite the code from assembly

```
v7 = *MK_FP(__GS__, 20);
v4 = __isoc99_sscanf(a1, "%d %d", &v3, &v2);
if ( v4 != 2 || v2 <= 1 || v2 > 4 )
  explode_bomb();
v5 = 9;
v6 = func4(9, v2);
if ( v6 != v3 )
  explode_bomb();
return *MK_FP(__GS__, 20) ^ v7;
```

With the v2 we get the condition to do nothing

Then use the loop to get v3 and v2 in code

```
int main()
{
    int v2; // [sp+18h] [bp-20h]@1
    int v3; // [sp+1Ch] [bp-1Ch]@1
    int v4; // [sp+20h] [bp-18h]@1
    int v5; // [sp+24h] [bp-14h]@5
    int v6; // [sp+28h] [bp-10h]@5
    int v7; // [sp+2Ch] [bp-Ch]@1

    for (int i = 0; i < 10; i++)
    {
        v2 = i;
        if (v2 <= 1 || v2 > 4)
        {
            continue;
        }
        v5 = 9;
        v6 = func4(9, v2);
        cout << v6 << " ";
        cout << v2 << endl;
    }
}
```

Then we get 3 password blow:

```
176 2
264 3
352 4
```

176 2

264 3

352 4

Phase 5:

In the task, we get the array in this

```
:0804D1BF                   db    8
:0804D1C0 ; int array_2704[]
:0804D1C0 array_2704        dd 0Ah                  ; DATA XREF: pha
:0804D1C4                   db    2
:0804D1C5                   db    0
:0804D1C6                   db    0
:0804D1C7                   db    0
:0804D1C8                   db  0Eh
:0804D1C9                   db    0
:0804D1CA                   db    0
:0804D1CB                   db    0
:0804D1CC                   db    7
:0804D1CD                   db    0
:0804D1CE                   db    0
:0804D1CF                   db    0
:0804D1D0                   db    8
:0804D1D1                   db    0
:0804D1D2                   db    0
:0804D1D3                   db    0
:0804D1D4                   db  0Ch
:0804D1D5                   db    0
:0804D1D6                   db    0
:0804D1D7                   db    0
:0804D1D8                   db  0Fh
:0804D1D9                   db    0
:0804D1DA                   db    0
:0804D1DB                   db    0
:0804D1DC                   db  0Bh
:0804D1DD                   db    0
:0804D1DE                   db    0
:0804D1DF                   db    0
:0804D1E0                   db    0
```

Then we re write the code with this assembly

```
int __cdecl phase_5(int a1)
{
  int v2; // [sp+14h] [bp-24h]@1
  int v3; // [sp+18h] [bp-20h]@1
  int v4; // [sp+1Ch] [bp-1Ch]@3
  int v5; // [sp+20h] [bp-18h]@3
  int v6; // [sp+24h] [bp-14h]@1
  int v7; // [sp+28h] [bp-10h]@3
  int v8; // [sp+2Ch] [bp-Ch]@1

  v8 = *MK_FP(__GS__, 20);
  v6 = __isoc99_sscanf(a1, "%d %d", &v2, &v3);
  if ( v6 <= 1 )
    explode_bomb();
  v2 &= 0xFu;
  v7 = v2;
  v4 = 0;
  v5 = 0;
  while ( v2 != 15 )
  {
    ++v4;
    v2 = array_2704[v2];
    v5 += v2;
  }
  if ( v4 != 15 || v5 != v3 )
    explode_bomb();
  return *MK_FP(__GS__, 20) ^ v8;
}
```

This is the code

```cpp
// Example program
#include <iostream>
using namespace std;
int main()
{
    int v2; // [sp+14h] [bp-24h]@1
    int v4; // [sp+1Ch] [bp-1Ch]@3
    int v5; // [sp+20h] [bp-18h]@3
    int v7; // [sp+28h] [bp-10h]@3
    int v8; // [sp+2Ch] [bp-Ch]@1
    int array_2704[16] = {10, 2, 14, 7, 8, 12, 15, 11, 0, 4, 1, 13, 3, 9, 6, 5};
    for (int i = 0; i <= 16; ++i)
    {
        cout << "case " << i << ": \n";
        v2 = i;
        v2 &= 0xF;
        v7 = v2;
        v4 = 0;
        v5 = 0;
        while (v2 != 15)
        {
            ++v4;
            v2 = array_2704[v2];
            v5 += v2;
            cout << v2 << "\t" << v5 << "\n";
        }
        if (v4 != 15)
            cout << "false\n\n";
        else
            cout << "true\n\n";
    }
}
```

We will get 16 case but we need to get the v5 of the round, so that we need to run from 0 to 16 to get the data

With the other case is the same as 6 is false

```
case 6:
15      15
false
```

Only the case 5 is true and the v2 is 5

```
case 5:
12      12
3       15
7       22
11      33
13      46
9       55
4       59
8       67
0       67
10      77
1       78
2       80
14      94
6       100
15      115
true
```

Then the v5 is maybe 15 or 115. After try 2 number the password of this phase is:

5 115

Test in Linux: