mud lecture

[Rude Lecture] 10. Relocation table

why do you ask
2014. 2. 18. 3:32

add neighbor

Starting today, we will design a program that can actually be used.
I'm going to try to cover some of the problems that arise when programming.
It's a crap lecture. I want to deal with a lot of things other than EUD.

First of all, since I briefly mentioned what trigger programming is,
Starting today, we are going to create a program that makes trigger programming easier.
Simply put, it's a compiler.
Assembler in trigger programming.

In fact, even if it is a programming tool for EUD, it is not much different from just a tool.

1. Relocation table

In fact, the trigger programming course ended in the 8th.
That's all we've covered in Lesson 8. From now on, I will only use the contents of the 8th lecture like crazy.

From the 9th, how to **apply** trigger programming ,
How can the trigger **easier** if you can weave **my own idea of** the process to introduce.

(Actually, the EUD course ended in the first lecture...)

The relocation table is the highlight of this course.
Is the term difficult? It's actually not difficult. I thought a relocation table would be better than a r elocation table, so I just wrote it in English.

First, let's find out what a relocation table is.

whyask37's blog

0

y (0x0058DC60 + 8 + 320 + 32 * 0 + 16 , Add , 1), SetMemory (0x0058DC60 + 8 + 320 + 32 * 0 + 20 , Add , 1), SetDeaths (Player1 , Add , 1 , 2), PreserveTrigger ()]) # trigger b B trigB = Trigger (players = [], conditions = [Deaths (Player1 , Exactly , 320 , 2)], Actions = [SetMemory (0x0058DC60 + 2408 + 4 , SetTo , 0xFFAE5D73), SetMemory (0x51A28C + 4 , SetTo , 0x51A28C), SetMemory (0x51A28C + 8 , SetTo , 0xFFAE5D73),]) trigAHeader = struct . pack ('<LL' , 0x0058DC60 + 2408 , 0x0058DC60 + 2408) trigBHeader = struct . pack ('<LL' , 0x0058DC60 , 0x0058DC60) data = b '' . join ([trigAHeader , bytes (trigA), trigBHeader , bytes (trigB)]) open ('payload.trg' , 'wb') . write (data)

It's like this. Did I mention that this trigger is placed in the MRGN section?
That is, **this trigger goes up at an offset of 0058DC60** .

If the star patch is 1.16.2 and the offset of the location table is 00000000, how will the above trigger change?
More generally, given the offset locoffset of the location table, how does the above trigger change?

Looking back on what I did in the previous lecture, I think the code might change as follows.

from trgformat import * import struct locoffset = 0x00000000 # offset of MRGN section # MRGN data triggers = [] # trigger A trigA = Trigger (players = [], conditions = [], actions = [SetMemory (0x0058D740 , SetTo , 0x00000000), SetMemory (**locoffset + 8 + 320 + 32 * 0 + 16** , Add , 1), SetMemory (**locoffset + 8 + 320 + 32 * 0 + 20** , Add , 1), SetDeaths (Player1 , Add , 1 , 2), PreserveTrigger ()]) # trigger b B trigB = Trigger (players = [], conditions = [Deaths (Player1 , Exactly , 320 , 2)], actions = [SetMemory (**locoffset + 2408 + 4** , SetTo , 0xFFAE5D73), SetMemory (0x51A28C + 4 , SetTo , 0x51A28C), SetMemory (0x51A28C + 8 , SetTo , 0xFFAE5D73),]) trigAHeader = struct . pack ('<LL' , **locoffset + 2408** , **locoffset + 2408**) trigBHeader = struct . pack ('<LL' , **locoffset** , **locoffset**) data = b '' . join ([trigAHeader , bytes (trigA), trigBHeader , bytes (trigB)]) open ('payload.trg' , 'wb') . write (data)

As in the code above, if the memory offset at which the trigger is raised is different,
There are values that need to be changed in response to the memory offset the trigger goes up.
At this time, there will be values that need to be changed like this,
The table in which the offsets are recorded within the trigger of these values is called a relocation table.

Values corresponding to the player side of EUD must be added by locoffset / 4 (because player is *4 in EUD)
Values corresponding to number side of EUD or prev and next pointer side of trigger must be added by locoffset.

whyask37's blog

0

Now let's generalize.
Let the trigger's origin be the starting address at which the trigger is loaded when the trigger is l oaded into memory.
At this time, when the actual origin of the trigger is different from the original origin by value (tha t is, when the trigger is loaded at address origin + value)
- Player relocation table to be added by value / 4 (let's call it PRT)
- Offset relocation table to be added by value (let's call it ORT)

There is.

The idea for the relocation table came from the PE format. (.reloc section)
If you are confident in programming, search ASLR.

[Correction 1] I deleted the whole concept of modules because of my lack of programming skill s. It is very difficult to make a tool.

#IT Computer

0

why do you ask

This is whyask37's blog.

add neighbor

this blog mud lecture Category article

whyask37's blog

0

2013. 10. 19.

11

5. SFmpq (ShadowFlare's MPQ Library) and examples

2013. 9. 11.

One

[Middle Lesson] 13. Trigger Programming - TRIG-MRGN Loop

2014. 2. 24.

0

4. scenario.chk

2013. 9. 10.

0

[Rude Lecture] 2. Substitution between death, plus

2014. 1. 19.

One

back to top

blog market
특수목으로 만든 무전력 스피커

View in PC version