Subject: Re: Syncing or changing BuildingGameObj 'IsDetroyed' state for clients
Posted by iRANian on Sun, 04 May 2014 09:25:09 GMT
View Forum Message <> Reply to Message

To find the function epilogue to patch open Renegade and attach OllyDbg, make sure Renegade is already in the main menu. in OllyDbg go to 0x006843E0 then follow the jump at the location. The new scripts 4.1 uses SSE heavily so the instructions for functions look kinda weird. Scroll up to find this kind of pattern at a function prologue:

ORIGINAL RENEGADE CODE AS EXAMPLE, THE TT CODE LOOKS DIFFERENT BUT ACTS THE SAME:

```
mov     al, [esp+20h+var_11]
test    al, al
jz      short loc_68431E
mov     al, [ebx+770h]
test    al, al
jnz     short loc_68431E
mov     edx, [ebx-8]
lea     ecx, [ebx-8]
call    dword ptr [edx+94h]
```

```
pop     edi
pop     esi
pop     ebx
add     esp, 14h
retn    4
```

All you really need is to find the check with 0x770 and a virtual function call to edx+0x94. Patch the epilogue so offset 0x770 is given the content of the byte stack variable that is tested for zero before the test for 0x770 being tested for zero in the code above. In this case:

```
mov     al, [esp+20h+var_11]
test    al, al
```

Happens before:

```
jz      short loc_68431E
mov     al, [ebx+770h]
```

So the epilogue needs to be patched so that offset 0x770 is updated with the content of [esp+20h+var_11].

Use OllyDbg to patch the epilogue in memory. then select and copy the patched instructions and

save them somewhere. Undo these memory patches (select the patches and right click -> Undo Selection) then open bandtest.dll with a hex editor, then find the epilogue in of the function in your hex editor by searching for the instruction bytes for the original epilogue (obviously make sure you find the correct one so check if there are multiple matches in the hex editor), replace the original epilogue instruction bytes with the instruction bytes have written down for your modified one. It might also be possible to just memory patch with OllyDbg and use the 'copy to executable' command.

Instruction bytes look like this:

```
64E016C9   8B7424 14       MOV ESI,DWORD PTR SS:[ESP+14]
64E016CD   8A46 0B         MOV AL,BYTE PTR DS:[ESI+B]
64E016D0   8886 70070000   MOV BYTE PTR DS:[ESI+770],AL
64E016D6   5F              POP EDI
64E016D7   5E              POP ESI
64E016D8   5B              POP EBX
64E016D9   83C4 14         ADD ESP,14
64E016DC   C2 0400         RETN 4
64E016DD   CC              INT3
```

The "8B7424 14" on the first line are 4 bytes for the instruction on the right of the line, "8A46 0B" on the second line are 3 bytes for the instruction on the right of that line etc.

Once done load up the game with the hex edited bandtest.dll and find the epilogue for the BuildingClass::Import_Rare() function again and check if your hex edits match the patched code your wrote down earlier, the code patches you applied with a hex editor.

I've attached a patched bandtest.dll, I have NOT checked if it works correctly with building revival. If the game crashes during startup or just after joining a server the file is incompatible with your version of 4.1.

File Attachments

1) bandtest.zip, downloaded 100 times