Bunnyhop Hack

The next hack we will implement in this seminar is a simple Bunnyhop hack. Bunnyhopping refers to the act of quickly, repeatedly jumping. With this in mind, the bunnyhop hack should, when enabled, make the player jump as soon as he lands on the ground. In Quake-based games such as CSGO, bunnyhopping allows you to move faster than normal, and, more importantly, looks cool.

Similar to last week, you will not be writing the full hack from scratch; you will only be asked to reverse-engineer the game using Cheat Engine, ReClass, and debuggers. We will tackle fully writing hacks in C beginning next week. Again, you will be expected to enter in only static offsets to implement the hack, but the bunnyhop hack requires more complicated memory accesses, including pointers and offsets you learned about this week.

Conceptually, to make the hack work we need two parts:

- A way to make ourselves jump (think back to +attack; there is also a +jump)
- A way to detect whether our player is on the ground.

The first part is very similar to last week's homework. As for the second part, there is a value in client_panorama.dll named m_fFlags which stores a bitmap holding your player's status flags. Examples of these flags include crouching, being in water, and importantly, being on the ground. Because this value is a field inside the game's base entity class (CBaseEntity), every entity in the game has its own copy of m_fFlags. You will need to locate your player entity's copy of this value along with a way to do so reliably using only static addresses.

Similar to the CrosshairID value we saw in class, this value is a part of your local player's data. This means that it is at a static offset from your local player's base address, and we saw in class that there is a static pointer to that base address. Hence, your task is to locate that static pointer as well as the struct offset for m_fFlags.

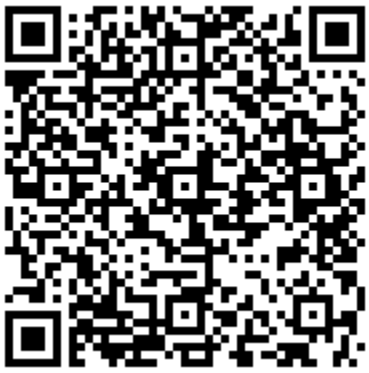
Once you've found the values, enter them into the demo program and try it out!

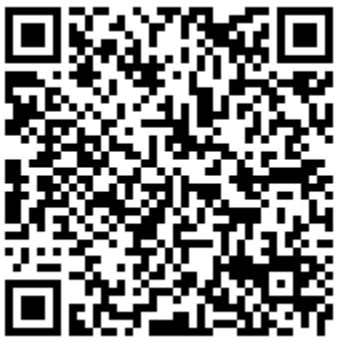
Due 2/11

Tips/tricks:

- You can use the command +duck to make yourself crouch. Don't forget about cl_pdump.
- Cheat Engine displays static addresses in green in the Found Address List.
- If you're curious, there are public and leaked Source Engine sources available online:
  https://github.com/ValveSoftware/source-sdk-2013/blob/master/sp/src/public/const.h
  https://github.com/VSES/SourceEngine2007
  These are a great aid for reverse engineers 😊

Hints (scan to view):





If you’re stuck, ask on Slack for help.