Note:

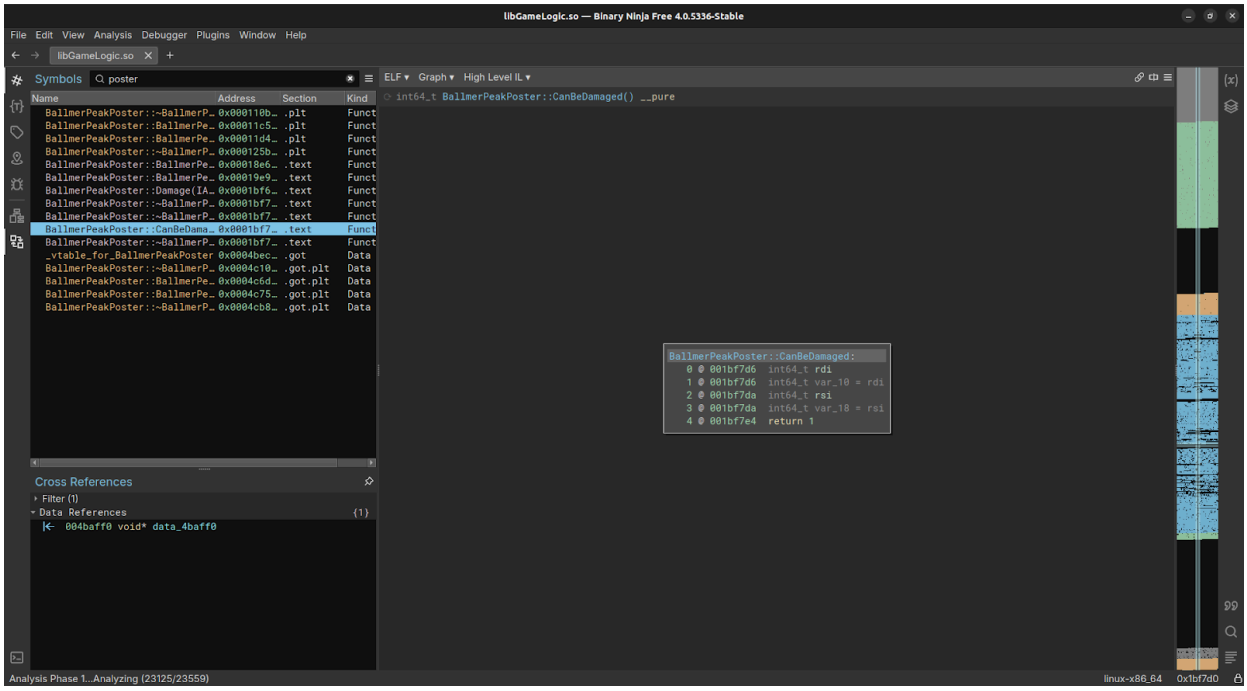
I took help from the liveoverflow youtube channel.

Project 1

- I initially tried flying high above the ground to avoid the bears, but this caused me to cross the boundary and end the mission.
- Next, I climbed a tree, thinking it would be safe. However, angry bears appeared and shot me with AK rifles when there were about 1:30 minutes left on the timer.
- When I experimented with the teleport (tp) command, I discovered that bears couldn't shoot below the ground. So, I teleported to the chest, started the timer, then used the "under" command to go underground and wait for the time to run out. The "under" command sets a flag and moves the player to a position 150 units below the current level. I applied this at every world Tick, but this caused the player to vibrate a lot. The player would fall due to gravity at each Tick, and the position reset caused occasional out-of-bounds issues. To solve this, I set the player's velocity to (0, 0, 0) and waited for 5 minutes.
- After logging out and back in, I returned to the chest and claimed my flag.

Project 2

- From the hint in the problem statement, I inferred that the eggs are objects in the set `((*ClientWorld)GameWorld).m_actors`, which contains `IActors`. I cast their `m_object` to `Actor` to retrieve the actors in the World.
- Upon inspection, I noticed that none of them had a display name, but they did have a `m_blueprintName`, which helped me identify the eggs.
- I then used a vector to store the locations of the eggs and wrote a script that allowed me to teleport to an egg's location by typing `egg <egg-number>` in the chat box.
- Using this method, I was able to find nine eggs. However, I discovered one more actor named `BallmerPeakEgg`, but no egg was present at that location. There was another actor called `BallmerPeakPoster`, which turned out to be an xkcd comic. When I decompiled the code, I saw that `CanBeDamaged()` always returns true. So, I purchased a CowBoyCoder and shot the Poster. This caused the final egg, the tenth one, to appear.


libGameLogic.so — Binary Ninja Free 4.0.5336-Stable
File Edit View Analysis Debugger Plugins Window Help
libGameLogic.so
Symbols
poster
Name Address Section Kind
BallmerPeakPoster::~BallmerP… 0x000110b… .plt Funct
BallmerPeakPoster::BallmerPe… 0x00011c5… .plt Funct
BallmerPeakPoster::BallmerPe… 0x00011d4… .plt Funct
BallmerPeakPoster::~BallmerP… 0x000125b… .plt Funct
BallmerPeakPoster::BallmerPe… 0x00018e6… .text Funct
BallmerPeakPoster::BallmerPe… 0x00019e9… .text Funct
BallmerPeakPoster::Damage(IA… 0x0001bf6… .text Funct
BallmerPeakPoster::~BallmerP… 0x0001bf7… .text Funct
BallmerPeakPoster::~BallmerP… 0x0001bf7… .text Funct
BallmerPeakPoster::CanBeDama… 0x0001bf7… .text Funct
BallmerPeakPoster::~BallmerP… 0x0001bf7… .text Funct
_vtable_for_BallmerPeakPoster 0x0004bec… .got Data
BallmerPeakPoster::~BallmerP… 0x0004c10… .got.plt Data
BallmerPeakPoster::BallmerPe… 0x0004c6d… .got.plt Data
BallmerPeakPoster::BallmerPe… 0x0004c75… .got.plt Data
BallmerPeakPoster::~BallmerP… 0x0004cb8… .got.plt Data
Cross References
Filter (1)
Data References {1}
004baff0 void* data_4baff0
ELF Graph High Level IL
int64_t BallmerPeakPoster::CanBeDamaged() __pure
BallmerPeakPoster::CanBeDamaged:
0 @ 001bf7d6 int64_t rdi
1 @ 001bf7d6 int64_t var_10 = rdi
2 @ 001bf7da int64_t rsi
3 @ 001bf7da int64_t var_18 = rsi
4 @ 001bf7e4 return 1
Analysis Phase 1...Analyzing (23125/23559)
linux-x86_64
0x1bf7d0