Hw4 Writeup

Fire and Ice

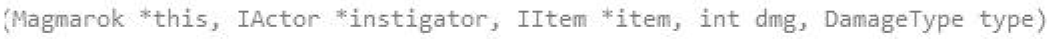
Team: Homeboys
Members: Jonathan Labrum, Josh Bowie, Glen Lauritsen, Nick Johnson

Goals Statement

Our goal, after examining the available flags, was to defeat Magmarok by discovering and exploiting an issue with the boss fight. By utilizing reversing tools we previously used in the course (IDA, Ghidra) we were able to successfully determine how to defeat Magmarok by utilizing an integer overflow with incoming healing to the boss.

Logic Analysis

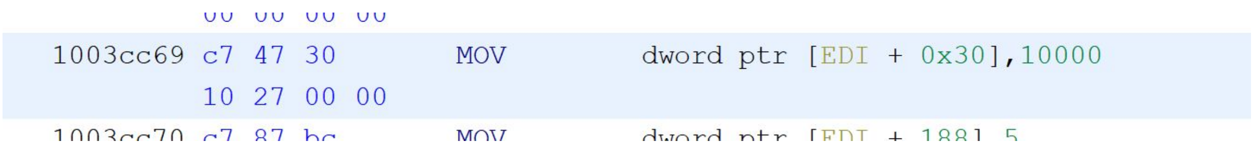
In order to figure out what happens behind the scenes with Magmarok's damage, we looked into the GameLogic.dll file on both IDA and Ghidra. They both had pros and cons. Within ghidra, we were able to see the decompiled functions but had the issue of not telling us function names or parameter names. However, we were able to get this information in IDA, so we cross referenced both in order to find and rename the functions that were important to us. In order to find the function that damaged Magmarok, we found the function Magmarok_Damage within IDA and then navigated to the function at that address in Ghidra. From there, we worked to understand what happened behind the scenes. In IDA, we were able to find the names of the parameters entered into the Magmarok_Damage function.



```
(Magmarok *this, IActor *instigator, IItem *item, int dmg, DamageType type)
```

We found during playing the game that fire spells healed him and other damage (guns and ice) would damage him. From there, we worked with the type parameter and it functioned as we thought, as different damage types caused Magmarok to be damaged in different ways. For example, if "dmg_type == 1", the damage would be recalculated and then negated, in essence healing him, and other damage types would be halved or doubled.

In addition, we followed the "this" parameter to see what class variables were accessed. We saw that "this + 0x30" was accessed quite a bit when the damage type was fire. We found Magmarok's constructor in IDA and saw the following instruction.

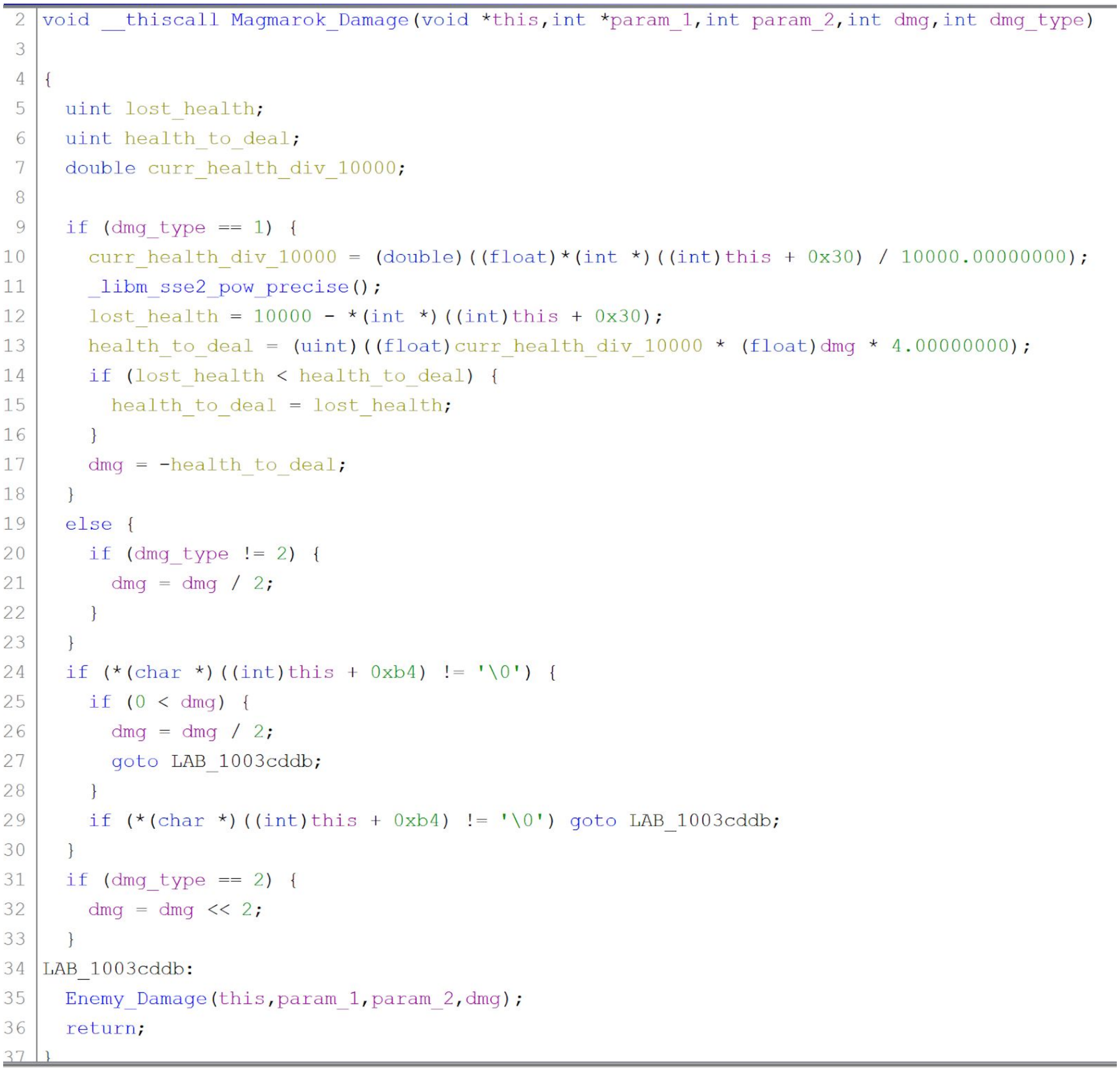


```
                 00 00 00 00
    1003cc69 c7 47 30        MOV        dword ptr [EDI + 0x30],10000
             10 27 00 00
    1003cc70 c7 87 bc        MOV        dword ptr [EDI + 188],5
```

This implies that this number is his health and it initialized to 10,000, which makes sense in the Magmarok_Damage function as this is accessed through various calculations. Upon renaming variables and some closed inspection, we found that while the damage parameter of the function is signed, all of the variables done in the fire calculation were unsigned. This meant that once his health went past 10,000, we could continue to heal him indefinitely, since we would not trigger the if statement preventing us to do so, as negative numbers interpreted as unsigned would be a large number, allowing us to heal him indefinitely.

Also, we found earlier that upon Magmarok falling to half health, we would regain half of his health, however instead of being set to 10,000, his health was incremented by 5,000. This function was the key to getting his health past the max, as we cannot do so through damaging him. We could increase it if it is already above 10,000, but not past the actual limit of 10,000 through the unsigned variable check.

The combination of these two facts means that if we can get Magmarok to heal himself past 10,000, we could heal him indefinitely until his health overflows to negative, killing him. We worked through the source code of the Magmarok_Damage function and came up with this translation.



```
void __thiscall Magmarok_Damage(void *this,int *param_1,int param_2,int dmg,int dmg_type)

{
  uint lost_health;
  uint health_to_deal;
  double curr_health_div_10000;

  if (dmg_type == 1) {
    curr_health_div_10000 = (double)((float)*(int *)((int)this + 0x30) / 10000.00000000);
    _libm_sse2_pow_precise();
    lost_health = 10000 - *(int *)((int)this + 0x30);
    health_to_deal = (uint)((float)curr_health_div_10000 * (float)dmg * 4.00000000);
    if (lost_health < health_to_deal) {
      health_to_deal = lost_health;
    }
    dmg = -health_to_deal;
  }
  else {
    if (dmg_type != 2) {
      dmg = dmg / 2;
    }
  }
  if (*(char *)((int)this + 0xb4) != '\0') {
    if (0 < dmg) {
      dmg = dmg / 2;
      goto LAB_1003cddb;
    }
    if (*(char *)((int)this + 0xb4) != '\0') goto LAB_1003cddb;
  }
  if (dmg_type == 2) {
    dmg = dmg << 2;
  }
LAB_1003cddb:
  Enemy_Damage(this,param_1,param_2,dmg);
  return;
}
```

Defeating the Magmarok

We first damaged him with ice until his health was at half then the Magmarok recovered itself. Even with all four of us hitting it with ice, its health never went below half. Once he fell below half, he began his healing animation. Next we kept shooting it with fire to overfill his health. It was hard to tell if it was working, but when we hit it with damaging attacks after some fire hits, its health did not go down. We kept hitting it with fire until its health overflowed and the Magmarok went down.