|  |  |
| --- | --- |
| 팀명 | 썽조 보유팀 |
| 문제명 | Duckbin (1) |
| 문제 풀이과정 작성(스크린샷 필수) | |
| Duck desktop을 만들어놓은 문제이다.  .NET이기 때문에 dnSpy를 통해 열어 디버깅을 진행하였다.  MainGame.Init() – ModSupport.LoadMods() – mod2.Init()    해당 코드에서 GooseAutumn.dll 파일을 열고 있는 부분을 발견하였다.  ModEntryPoint – CheckFlag  함수의 이름을 통해 해당 함수를 통과하면 될 것으로 예상하고 분석을 시작하였다.  내부에 VM을 구현하여 커스텀 스택 및 커스텀 opcode를 통해 사용자의 입력값을 검사한다. 해당 데스크탑 파일 상에서는 키보드를 후킹하여 값을 입력받았기 때문에 해당 코드를 따로 옮겨 분석을 진행하였다.    코드가 길지 않고, 중간에 함정이 많이 섞여 있어 디버깅을 진행하며 opcode를 분석하였다.  대략적인 루틴은 다음과 같다.   1. 사용자의 입력 값과 77과 한 바이트 씩 XOR 2. 별개의 XOR table을 통해 1번의 결과값과 한 바이트 씩 XOR 3. ADD, AND, SUB 연산을 조합하여 2번의 결과값을 특정 결과값과 한 바이트 씩 비교   위 루틴을 모두 분석한 후, z3를 이용하여 플래그를 얻어내었다.   |  | | --- | | from z3 import \*  S = [BitVec('flag%i' % i, 32) for i in range(41)]  s = Solver()  stage2\_xor\_table = [      140, 98, 22, 87, 105, 114, 185, 15, 86, 46, 30, 211, 207, 174, 147, 232, 231, 28, 60, 127, 13, 67, 37, 106, 109, 70, 230, 205, 208, 56, 29, 138, 140, 229, 200, 244, 13, 4, 179, 122, 48  ]  stage3\_result\_table = [      44, 54, 122, 47, 90, 13, 99, 9, 244, 143, 140, 52, 27, 88, 154, 135, 77, 78, 159, 9, 233, 143, 179, 194, 18, 25, 225, 203, 71, 20, 11, 101, 71, 11, 74, 243, 177, 60, 204  ]  stage3\_add\_add\_idx = [      0, 2, 10, 11, 13, 20, 21, 23, 27, 31, 36, 37  ]  stage3\_sub\_add\_idx = [      14, 15, 16, 26, 28, 38, 1  ]  stage3\_sub\_sub\_idx = [      5, 6, 7, 8, 12, 17, 18, 22, 25, 29, 32, 33, 35  ]  stage3\_add\_sub\_idx = [      3, 4, 9, 19, 24, 30, 34  ]  s.add(S[0] == ord('c'))  s.add(S[1] == ord('c'))  s.add(S[2] == ord('e'))  s.add(S[3] == ord('2'))  s.add(S[4] == ord('0'))  s.add(S[5] == ord('2'))  s.add(S[6] == ord('3'))  s.add(S[7] == ord('{'))  for i in range(2, 41):      # print(i)      if i-2 in stage3\_add\_add\_idx:          s.add(((((S[i-2]^77^stage2\_xor\_table[i-2]) + (S[i-1]^77^stage2\_xor\_table[i-1])) & 0xFF) + (S[i]^77^stage2\_xor\_table[i])) & 0xFF == stage3\_result\_table[i-2])      elif i-2 in stage3\_add\_sub\_idx:          s.add(((((S[i-2]^77^stage2\_xor\_table[i-2]) + (S[i-1]^77^stage2\_xor\_table[i-1])) & 0xFF) - (S[i]^77^stage2\_xor\_table[i])) & 0xFF == stage3\_result\_table[i-2])      elif i-2 in stage3\_sub\_add\_idx:          s.add(((((S[i-2]^77^stage2\_xor\_table[i-2]) - (S[i-1]^77^stage2\_xor\_table[i-1])) & 0xFF) + (S[i]^77^stage2\_xor\_table[i])) & 0xFF == stage3\_result\_table[i-2])      else:          s.add(((((S[i-2]^77^stage2\_xor\_table[i-2]) - (S[i-1]^77^stage2\_xor\_table[i-1])) & 0xFF) - (S[i]^77^stage2\_xor\_table[i])) & 0xFF == stage3\_result\_table[i-2])  if s.check() == sat:      m = s.model()      w = ''      for i in range(41):          w += chr(m[S[i]].as\_long())      print(w)  cce2023{0fa609cd45b11ee0cc26cb0d9bba5927} | | |

|  |  |
| --- | --- |
| 팀명 |  |
| 문제명 |  |
| 문제 풀이과정 작성(스크린샷 필수) | |
|  | |