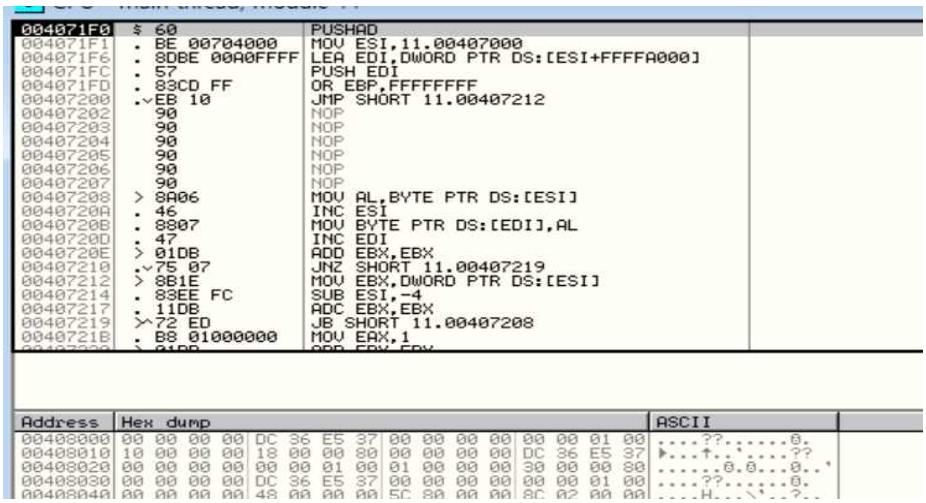
CodeEngn Basic RCE L11
: OEP를 찾으시오. Ex) 00401000 / Stolenbyte 를 찾으시오.

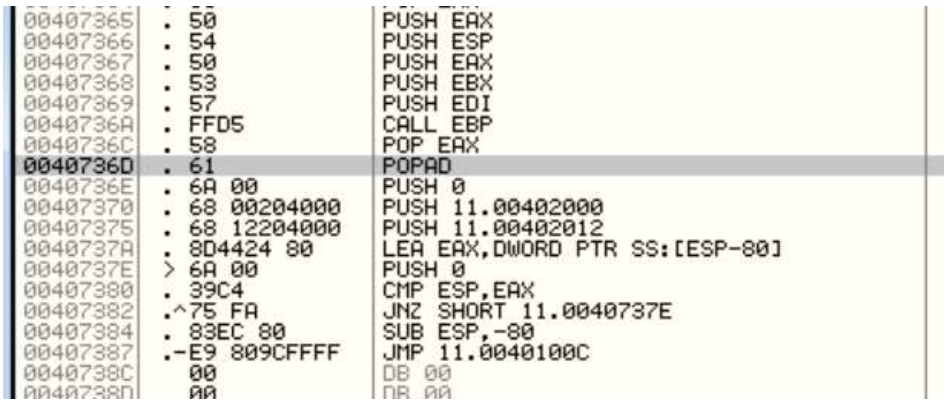
Ex) FF35CA204000E84D000000 정답인증은 OEP+ Stolenbyte
Ex) 00401000FF35CA204000E84D000000

1. OEP 구하기



올리디버거에 올려보았더니 바로 pushad가 보인다.

OEP코드를 찾기 위해 PUSHAD의 짝꿍 POPAD를 찾아준다.



조금만 내려봤더니 POPAD 명령어가 존재해서, 앞뒤 부분의 코드를 살펴보면서 OEP로 점프하는 명령문을 찾아보겠다.

해당 주소로 점프하는 명령어는 JMP와 JNZ가 있고,
이 둘 중 하나가 OEP코드가 있는 곳으로 이동하라는 명령을 지시한다고 추측된다.
그럼 반복되지 않는 부분이 OEP코드로 이동하는 부분이기 때문에,
각각의 명령어를 분석해 볼 필요가 있다.

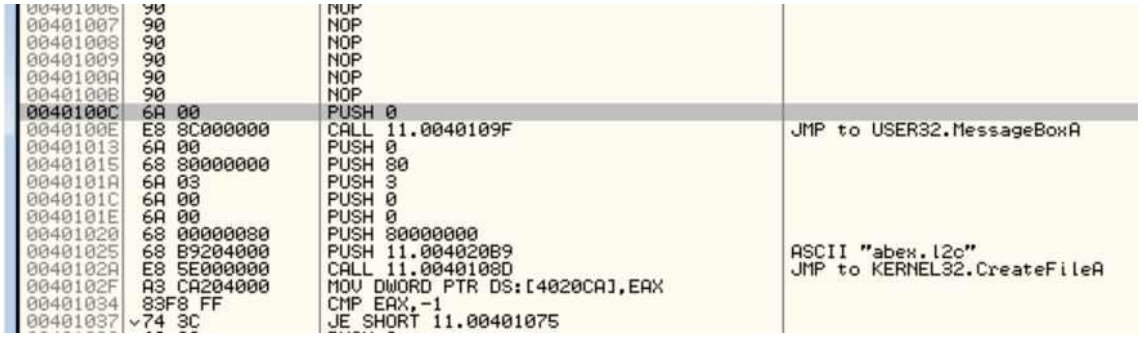
JNZ = JUMP NOT ZERO
0이 아니면 점프!이기 때문에 JNZ 바로 위 명령어인
CMP ESP, EAX는 ESP-EAX를 명령하고,
바로 밑에 JNZ는 ESP와 EAX가 같아야 이후 명령어인 SUB ESP, -80을 수행할 수 있으니깐
JNZ명령어에서 ESP와 EAX가 같지 않아서 계속 0040737E PUSH 0 으로 이동한다는 점을 알 수 있다.

따라서 OEP 코드가 있는 곳은 그 밑에 있는 JMP 명령어에 있는 0040100C이라는 것을 알 수 있다.

2. Stolenbyte 구하기

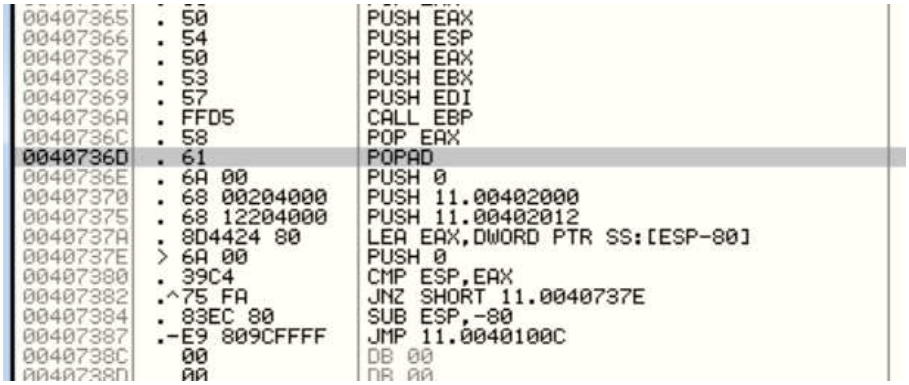
stolenbyte는 OEP로 이동하기 전에 PUSH 되는 것들입니다.
아까 OEP코드를 구할 때 총 3개의 인자가 푸시된다는 것을 알 수 있었는데, 이것이 훔쳐진 바이트를 구하는데 핵심되니 기억해두자.

각각 PUSH인자와 점프 명령어에 BP를 걸어주고 F9를 눌러 실행 시켜주었더니 아스키 코드 문자열이 보였고, 위에서 구한 OEP주소로 이동했다.



PUSH 0 위로 NOP으로 쭉 채워진 것을 확인할 수 있는데,
그 밑에 호출하는 함수 메세지 박스는 총 4개의 파라미터를 필요로 한다.

그래서 3개의 인자가 부족하고,
이전에 OEP로 이동하기 전 3개의 PUSH된 부분이 stolenbyte라는 것을 알 수 있다.



6A0068002040006812204000 이라는 것을 알 수 있다.

따라서 11번의 키는 0040100C6A0068002040006812204000 이다.