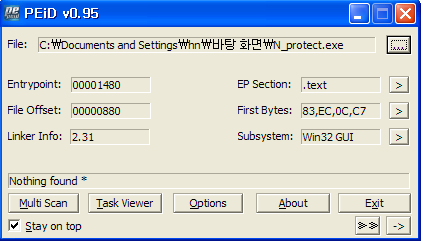
4. 악성코드 backdoor 상세 분석(N\_protect.exe)

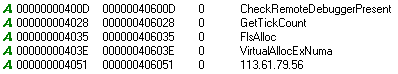
4.1 정적/동적 분석

4.1.1 패킹 확인

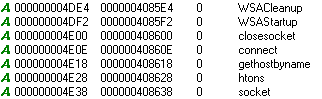


악성코드인 N\_protect.exe의 패킹여부 확인 결과, 패킹이 되어 있지 않을 것을 확인할 수 있었습니다.

4.1.2 문자열 확인

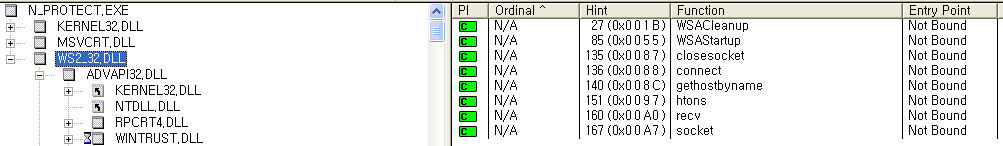


CheckRemoteDebuggerPresent로 안티디버깅하며, 113.61.79.56는 해커 IP라고 추측할 수 있습니다.

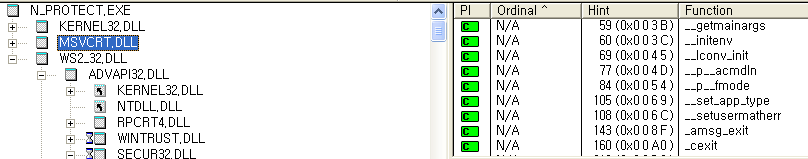


WSACleanup과 WSAStartup, connect, socket함수를 보니 소켓을 생성해서 통신하는 것을 예상할 수 있습니다.

4.1.3 dll 구조 확인



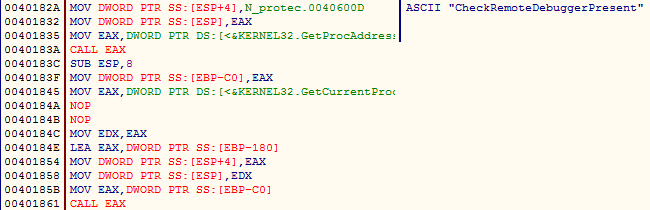
웹 소켓 함수들이 필요한 것을 보니 소켓통신 하는 것을 확인할 수 있습니다.



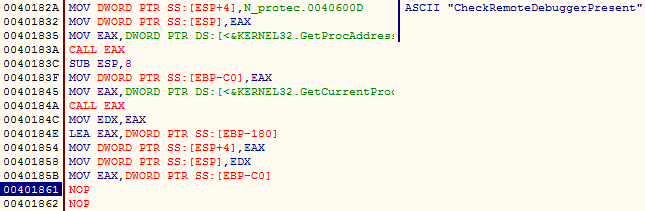
getmainargs은 명령줄 구문 분석을 호출하고 포인터를 통해 다시 main()으로 인수를 복사하는 함수입니다. 이것을 통해 main가 포인터를 이용하여 숨겨 있을 것이라고 예상할 수 있습니다.

4.2 상세 분석

4.2.1 안티디버깅 우회



CheckRemoteDebuggerPresent() 함수는 안티디버깅의 대표적인 함수입니다. 따라서 이것을 우회해야 끝까지 분석이 가능합니다. 우회를 위해서 함수가 실행되는 CALL EAX부분을 NOP으로 변경해야 합니다.



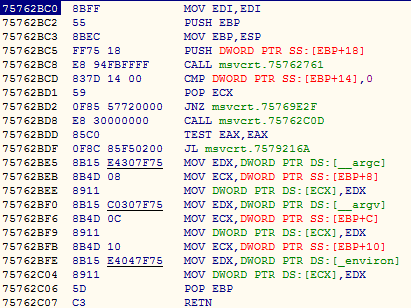
4.2.2 메인함수 찾기



모듈의 이름으로 getmainargs를 검색합니다.

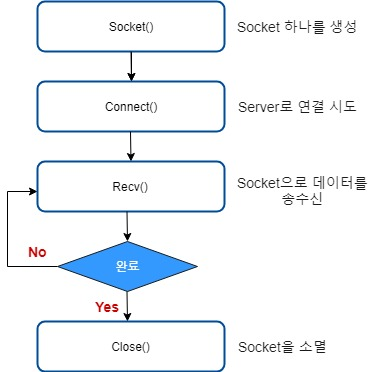


00404868의 함수를 실행합니다.



스택의 프롤로그, 에필로그 등 메인함수가 있는 것을 확인할 수 있습니다.

4.2.3 소켓 생성 및 실행



- WSAStartup

// 윈도우 소켓 초기화를 위해 WSAStarup을 호출합니다.





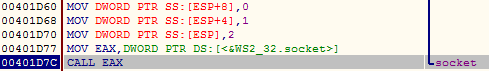
- socket

// 소켓을 생성합니다.

Family = AF\_INET : IPv4영역

Type =1 : SOCK\_STREAM(연결지향, 양방향의 TCP/IP기반의 통신을 위한 것)

Protocol = IPPROTO\_IP (주어진 소켓 종류의 디폴트 프로토콜 사용)





- gethostbyname

// 도메인으로 IP주소를 가져오는 함수입니다. 도메인을 이용해서 공격할 시 필요한 것입니다.







- htons

// 포트번호를 호스트 바이트 순서에서 네트워크 바이트 순서로 변환합니다. 115C는 십진수로 하면 4444를 의미합니다. 따라서 포트번호는 4444입니다.

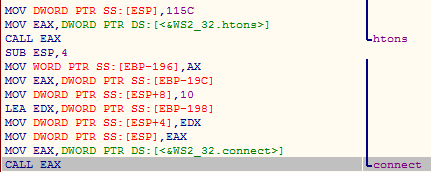






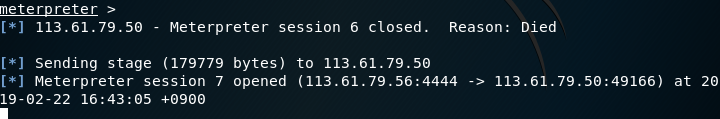
- connect

// 생성한 소켓을 통해 서버로 접속을 요청합니다.









connect()함수가 실행된다면, 세션이 열리는 것을 볼 수 있습니다. 이때부터 해커는 자신이 원하는 정보를 빼낼 수 있습니다.

- recv

// 접속되어 있는 소켓으로부터 들어오는 데이터를 수신하는 함수입니다.

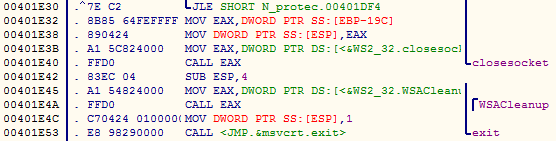






- closesocket

// 소켓을 종료할 때 사용하는 함수입니다.









closesocket()함수가 실행되면, 소켓이 닫히기 때문에 세션도 닫힙니다.

<별첨>

-메인함수 찾기



Search for -> name is all modules로 아까 dll구조에서 보았던 getmainargs를 검색합니다. 이후 msvcrt.\_\_getmainargs를 누릅니다.



둘 중에 위에 있는 CALL <JMP.&msvcrt.\_\_getmainargs>를 누릅니다.



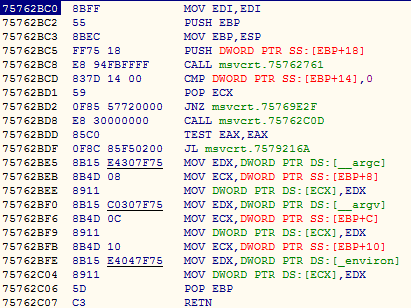
F7를 눌러 함수 안으로 이동합니다.



여기서 D8814000이므로 004081D8에 가서 뭐 있는지 확인하면



이 주소로 이동하면 ,,,~ 75762BC0



스택의 프롤로그, 에필로그 등 메인함수인 것을 확인할 수 있습니다.