2021

C Programming Project Portfolio

XOR Cipher Server

정준영(Haruster)

https://github.com/Haruster

01 프로젝트 소개

해당 프로젝트는 XOR 암호화 기법과 TCP / IP 서버를 결합시킨 통신 암호화 서버를 구현한 프로젝트라고 할 수 있습니다. 이러한 프로젝트를 제작하게 된 계기는 평소에 보안에 관심을 가지고 있었는데 C언어를 이용한 프로젝트를 기획하던 중 어떻게 하면 해당 프로젝트와 보안을 접목시킬 수 있을지 생각을 하게 되었고 암호화 중 CTF(해킹 방어 대회)에서 자주 쓰이는 방식인 XOR 암호화 기법과 TCP/IP Server를 이용해서 암호화 서버를 만들면 재밌을 것 같아서 해당 프로젝트를 진행하게 되었습니다.

02 제작하면서 느낀 점?

해당 프로젝트를 제작하면서 느낀 점은 암호화 서버를 직접 만들어보면서 암호화에 대해 조금이라도 더 배울 수 있었으며 아쉬운 점은 오류가 조금있다는 것이 아쉬웠다고 할 수 있습니다.

03 추후에 개선하고 싶은 점

추후에는 해당 XOR암호화 뿐만 아니라 RSA등의 암호화 함수를 적용시켜서 더 많은 암호화가 구현되게 만들고 싶으며 오류 사항도 수정하고 싶습니다.

11

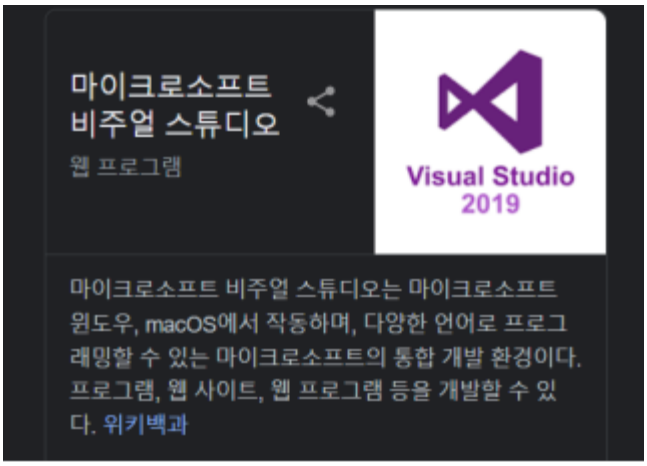
Tcp IP Server / Client 구현

계획

12

암호화 구현 및 decrypt 소스 구현

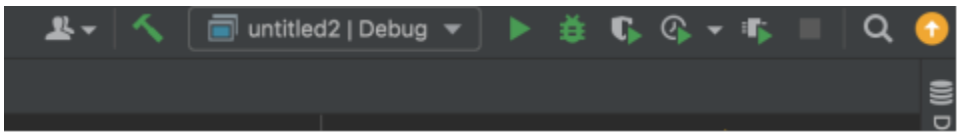
04 사용 방법



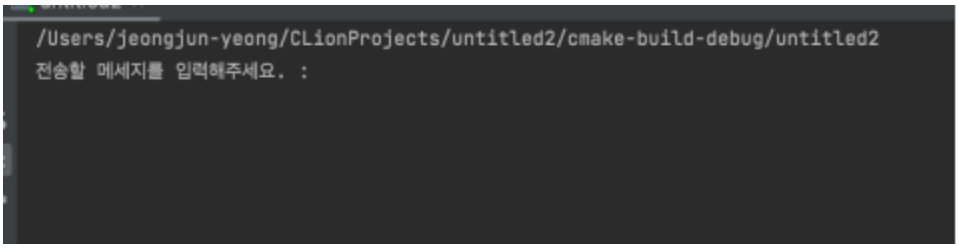
해당 소스코드를 사용하기 위해서는 C언어를 컴파일 할 수 있는 컴파일러 또는 IDE(통합 개발 환경)이 필요합니다. 다만 어느 특정한 IDE를 사용할 필요는 없으며 Visual Studio, Dev C++, CLion, ...등 자신한테 맞는 어느 IDE를 사용하여도 됩니다만 해당 문서에서는 CLion을 사용하였습니다.

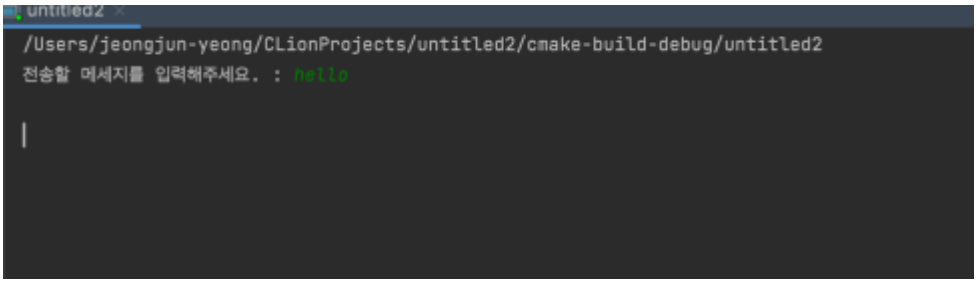
```c
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <sys/socket.h>

#define PORT 9000

char buffer[BUFSIZ];

char* XorCrypto(char* Data, char* Key, int DataLength, int KeyLength) {

    char* output = (char*)malloc( size: sizeof(char) * DataLength);

    for(int i = 0; i < DataLength; ++i) {

        output[i] = Data[i] ^ Key[i % KeyLength];

    }
```

IDE에 server.c의 소스코드를 복사 붙여넣기 한 후 Key 값을 자신이 원하는 값으로 설정해줍니다.

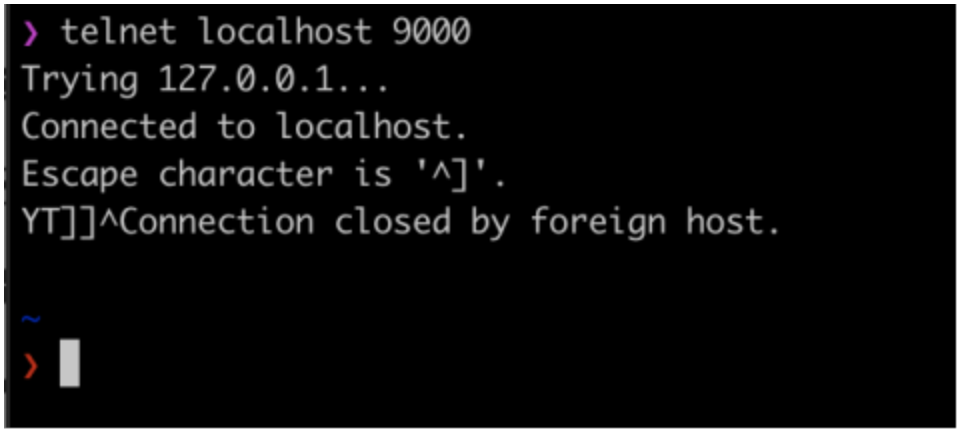


Key값을 설정하였다면 실행 버튼을 눌러서 해당 서버를 실행시킵니다.

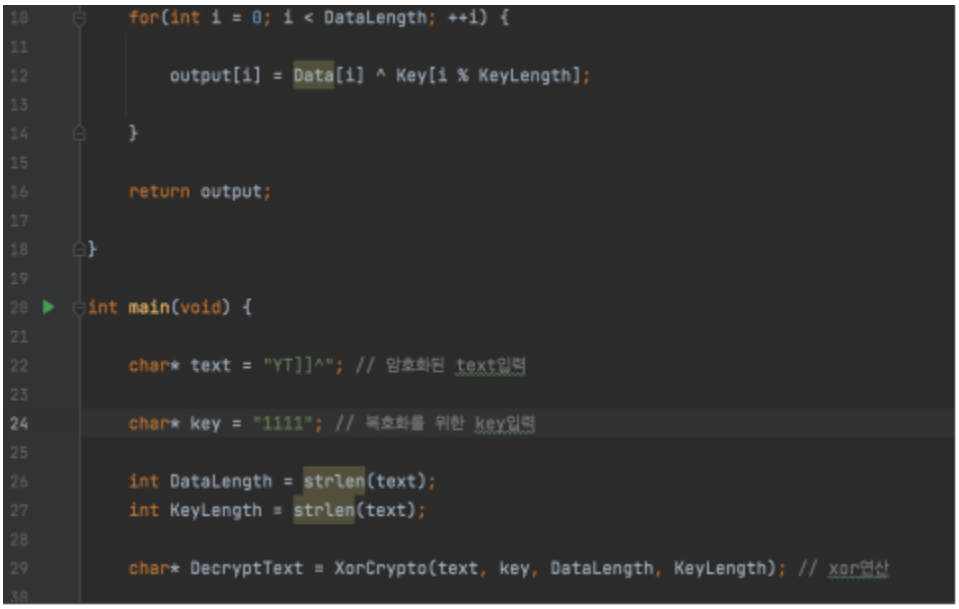




콘솔에 메시지가 뜨면 자신이 클라이언트에게 전송할 메세지를 입력해줍니다.

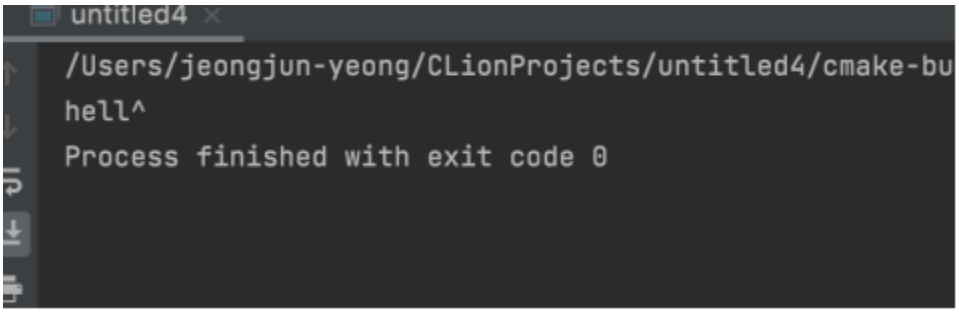


Telnet 또는 Client.c를 이용하여 서버가 보낸 내용을 확인하고 해당 메세지를 복사해둡니다.



```
    for(int i = 0; i < DataLength; ++i) {

        output[i] = Data[i] ^ Key[i % KeyLength];

    }

    return output;

}

int main(void) {

    char* text = "YT]]^"; // 암호화된 text입력

    char* key = "1111"; // 복호화를 위한 key입력

    int DataLength = strlen(text);
    int KeyLength = strlen(text);

    char* DecryptText = XorCrypto(text, key, DataLength, KeyLength); // xor연산
```

다음으로 decrypt_message.c의 소스코드를 복사하여 IDE에 붙여넣기 한 후 text, key를 설정한 후 이를 실행해줍니다.



복호화된 메세지를 확인해줍니다.(완벽하지 못한 부분이 존재합니다.)