1. Instruction pattern transformation

Việc thay đổi trình biên dịch sẽ tạo ra những biến đổi phức tạp hơn, và cũng ít phổ biến hơn nhưng mục tiêu là giống nhau và hiệu quả về mặt chống dịch ngược.

Ví dụ, Demo: Để biên dịch mã C++ có thể dùng nhiều trình biên dịch:

+ MinGW

+ Borland C ++

+ Dev C ++

+ Embarcadero C++ builder

+ Clang

+ Visual C ++

+ Intel C ++

+ Code block

Demo:

1. Dummy code insertion

Những đoạn mã giả, mã không liên quan đến mục tiêu của phần mềm có thể được thêm vào chương trình để làm cho chương trình khó đọc khi dịch ngược hơn, nhưng nó không ảnh hưởng đến logic hoặc kết quả của chương trình.

Ví dụ, Demo:

1. Opaque predicate insertion

If else trong mã là một biểu thức logic đúng hoặc sai. Đây là các nhánh có điều kiện - hoặc các câu lệnh if-then - trong đó kết quả không thể dễ dàng xác định được bằng phân tích thống kê. Việc chèn một hoặc nhiều câu lệnh rẽ nhánh không rõ ràng sẽ tạo ra những đoạn mã không cần thiết và cũng không bao giờ được thực thi nhưng gây khó hiểu cho người dịch ngược, đang cố gắng hiểu đầu ra và hành vi của chương trình.

1. Code transposition

Phương pháp này là sự sắp xếp lại quy trình xử lí các module và các nhánh trong mã mà không có ảnh hưởng rõ ràng đến hành vi của nó.