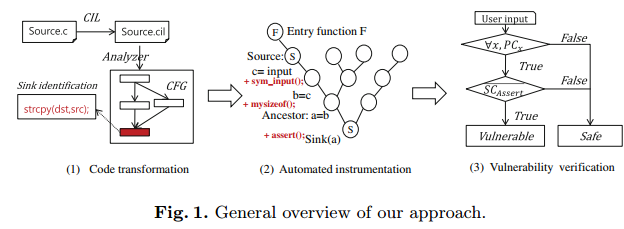
Kami di sini fokus pada kerentanan buffer overflow karena jenis kerentanan ini adalah penyebab utama terjadinya intensitas berbahaya seperti akses memori yang tidak valid, penolakan layanan (crach system) dan eksekusi kode yang sewenang-wenang.Alasan mengapa kita melakukan transformasi kode CIL adalah menyederhanakan struktur kode untuk analisis statik yang efisien.

We then identify the security sinks(potential vulnerable) to get potential vulnerable points. Then, we perform automatic program instrumentation and prepare the testing object for vulnerability verification. In the last phase of automated instrumentation, we verify each potential security sink to report vulnerabilities using concolic testing.



Dengan menggunakan API CIL yang tersedia, kita dapat meratakan struktur kode kompleks menjadi struktur yang sederhana (misalnya, semua konstruksi perulangan dikurangi menjadi bentuk tunggal, semua badan fungsi diberi pernyataan pengembalian eksplisit).

