گزارش نهایی فاز اول پروژه طراحی کامپایلر – MiniC مبهم‌سازی زبان

مقدمه

در این پروژه یک ابزار مبهم سازی (Obfuscator) برای زبان MiniC طراحی شده است که وظیفه

آن کاهش خوانایی و پیچیده‌سازی کد منبع بدون تغییر در عملکرد برنامه است .هدف از این کار، جلوگیری از مهندسی معکوس، افزایش امنیت و محافظت از منطق اصلی برنامه می‌باشد .در این گزارش، معماری کلی سیستم، پیاده‌سازی فنی، تکنیک‌های به کار رفته و مثال‌های قبل و بعد از .مبهم‌سازی بررسی می‌شوند

معماری کلی ابزار

ابزار طراحی‌شده از سه بخش اصلی تشکیل شده است

تجزیه کد ورودی با استفاده از ANTLR

اعمال تکنیک های مبهم سازی روی AST با کلاس ObfuscatorVisitor

تولید مجدد کد مبهم شده با کلاس CodeGenerator

این ساختار ماژولار، امکان توسعه و تست آسان‌تر را فراهم می‌سازد

main.py

این فایل نقطه‌ی ورود برنامه است و به ترتیب زیر عمل می‌کند

خواندن فایل ورودی MiniC

ایجاد Parser و Lexer با استفاده از گرامر MiniC

تولید AST با استفاده از parser

اعمال بازنویسی روی درخت نحوی

تولید کد مبهم شده از AST جدید

نوشتن خروجی در فایل نهایی

ObfuscatorVisitor

این کلاس از MiniCVisitor ارث بری می کند و روی گره های AST عملیات بازنویسی انجام می دهد

تغییر نام متغیرها و توابع به نام‌های بی‌معنا

اضافه کردن دستورات مرده به بدنه بلوک ها

بازنویسی تابع main با استفاده از while+switch-case

حفظ وابستگی معنایی و ساختاری کد در هنگام تغییر

.این بازنویسی با حفظ عملکرد برنامه، خوانایی آن را به‌شدت کاهش می‌دهد

تکنیک‌های مورد استفاده

تغییر نام متغیرها و توابع جهت از بین بردن معنای اولیه نام‌ها

افزودن متغیرها و دستورات بی‌استفاده به منظور گیج کردن تحلیل‌گر

شکستن ترتیب اجرای طبیعی و جایگزینی آن با ساختار Switch _ case

استفاده از عبارات معادل مانند a - (-b) به a + b

CodeGenerator

کدی را از AST تولید می کند

تولید تابع با پارامتر و بدنه :visitFunctionDefinition

ایجاد بلوک‌های کدی با تورفتگی مناسب :visitCompoundStatement

بازسازی عبارات محاسباتی :visitAdditiveExpression و visitAssignmentExpression

تولید فراخوانی توابع با پارامترهای لازم :visitFunctionCall

نمونه کد ورودی و خروجی

ساده به عنوان ورودی MiniC کد:

```c
int sum(int a, int b) {
    int result = a + b;
```

```
    return result;
}

int main() {
    int x = ۳;
    int y = ۴;
    int total = sum(x, y);
    printf("%d\n", total);
    return ۰;
}
```

نسخه مبهم‌شده همان کد:

```c
int fxz(int x۱, int x۲) {
    int a۳۹ = x۱ - (-x۲);
    int unused = ۱۲۳۴;
    return a۳۹;
}

int main() {
    int var۱ = ۳;
    int var۲ = ۴;
    int useless = ۰;
    int obf_result = ۰;
    int selector = ۱;
```

```
    while (selector > ٠) {
        switch(selector) {
            case ١:
                obf_result = fxz(var١, var٢);
                selector = ٢;
                break;
            case ٢:
                printf("٪d\n", obf_result);
                selector = ٠;
                break;
        }
    }
    return ٠;
}
```