Fix Lab 12: Code Analysis

Date Due: Tuesday, December 7, 2021 by 11:59 PM

Introduction

In this lab assignment, you will write your own analyzer. This assumes you have completed both Lab 0: Lab Setup Guide. For this lab you will need the ability to run Visual Studio 2019 Community Edition along with its Visual Studio Extensions Development plugins. You will also need to have completed the Git setup outlined in Lab 0 and the ability to push code changes to the remote repository on https://code.umd.edu.

Important Note: For each of the following questions, you will be adding a new set of projects within the same WebGoat .NET folder that is at the root of the codebase. For this lab, you can base the code off the main branch of your WebGoat .NET code repository. However, you will implement each phase within their own branches as specified by the questions in the phases. Not all phases have code submissions.

Phase 1: Microsoft Analyzer Tutorial

You will first go through creating the tutorial as outlined here: https://docs.microsoft.com/en-us/dotnet/csharp/roslyn-sdk/tutorials/how-to-write-csharp-analyzer-code-fix.

1. Implement the MakeConst Analyzer as specified in the tutorial and take the time to understand the following concepts:
    a. Microsoft.CodeAnalysis.DiagnosticAnalyzer
    b. Microsoft.CodeAnalysis.AnalysisContext
    c. DataFlowAnalysis, Symbols and createChangedDocument
    d. Microsoft.CodeAnalysis.Testing
2. Additionally read more about Semantic Analysis API here: https://docs.microsoft.com/en-us/dotnet/csharp/roslyn-sdk/get-started/semantic-analysis

Phase 2: Implement your own Analyzer

As you saw with the WebGoat .NET application and the SRP implementation, a weak random number generator was used for cryptographic operations. They allow for predictable sequences and can be detrimental to security.

1. Now attempt to create an analyzer that will check for any variables that use System.Random and suggest a fix to use System.Cryptography.RandomNumberGenerator instead.
2. Remember to commit and push your code under the Lab12_Phase2 branch.

Submission Criteria

Please submit a writeup with:

1. Your name, UMD email ID and Lab Number.
2. The answers provided in the format given at the top of this lab handout.

Please only submit a **Word document** or a **PDF**. This lab contains code submission with branches for each phase. Please ensure the commits submitted are accessible by the auto grader.