Opaque Predicates

Tyler Sedlar

December 25, 2018

1 Meaning

In computer programming, an opaque predicate is a predicate–an expression that evaluates to either "true" or "false"–for which the outcome is known by the programmer a priori, but which, for a variety of reasons, still needs to be evaluated at run time. Opaque predicates have been used as watermarks, as it will be identifiable in a program's executable. They can also be used to prevent an overzealous optimizer from optimizing away a portion of a program. Another use is in obfuscating the control or dataflow of a program to make reverse engineering harder.[1]

2 Resources

A very informative document by Dongpeng Xu, Jiang Ming, and Dinghao Wu can be found here. It explains the meaning, use cases, and different types of opaque predicates along with the namings for content associated with them.

3 Personal Notes/Summary

An opaque predicate is a predetermined branch used for watermarking code in obfuscation processes, which usually consists of checking a variable or parameter against a constant. An incorrect parameter value can cause a program to terminate early.

[1] `https://en.wikipedia.org/wiki/Opaque_predicate`