Breaking Abstractions and Unstructuring Data Structures

Christian Collberg Clark Thomborson Douglas Low

"**Mobile programs** are distributed in forms that are isomorphic to the original source code. Such codes are easy to decompile, and hence they increase the risk of malicious reverse engineering attacks…
Code obfuscation is a potent defence against reverse engineering. "

Reviewer: Hongying lai

Outline of the Paper

- What is obfuscation ?

  Obfuscation is a process that renders software unintelligible but still functional.

- Transformation quality

  - potency, resilience, stealth and cost.

- Technique of obfuscation

  - lexical transformation : modify the lexical structure of the program.
  - control transformation: alter control structures using opaque predicates.
  - data transformation : modify inheritance relations, ...

- Conclusion

In this presentation I will focus on data transformations.

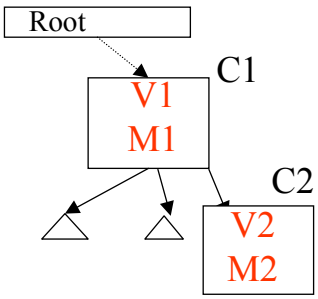
How to Obfuscate Data

- Class – modify inheritance.
- Procedural abstraction(Java methods)
  - convert a section of code into a different virtual machine;
  - inline some methods, and outline other methods;
  - clone methods.
- Variable – split built-in data types.
- Arrays – restructure
  - split an array into several sub-arrays;
  - merge two or more arrays into one array;
  - fold an array(increasing the number of dimensions);
  - flatten and array(decreasing the number of dimensions).

In this presentation I shall concentrate on modifying inheritance.

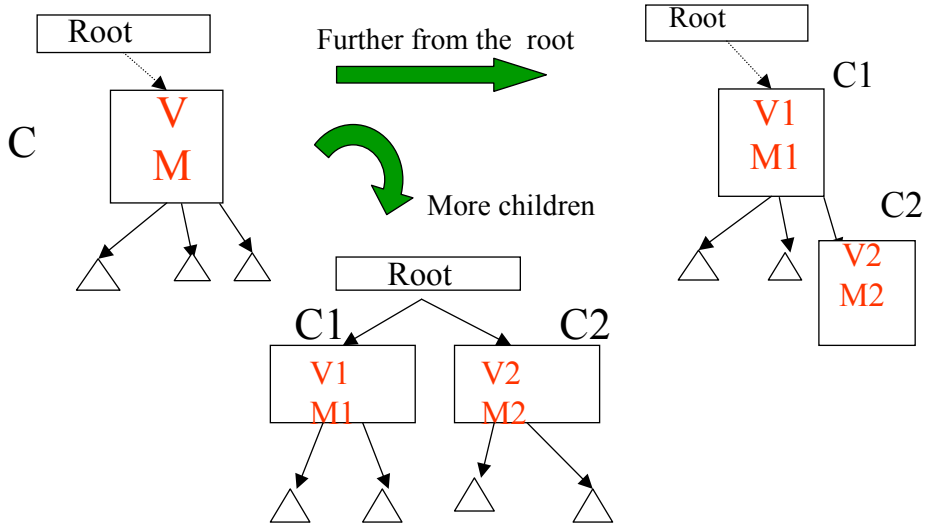
modify inheritance

- Review of stage-1 CS: what is a Java class
  - an encapsulation data(V) and control(M) .
  - an aggregation(C2 instance of type C1).
  - an inheritance (class C2 extends class C1) .

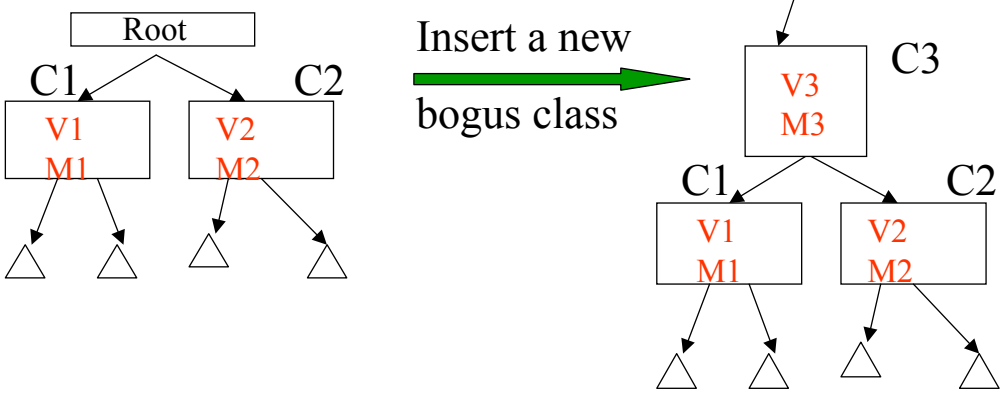


modify inheritance
(factoring classes)

- the complexity of a class grows with
  - its depth in the inheritance hierarchy.
  - the number of its direct descendants.



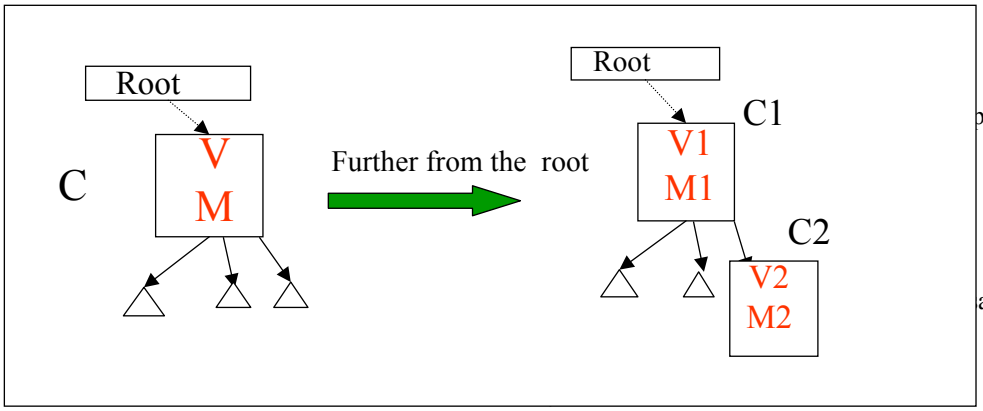
modify inheritance
(false refactoring classes)



example

```
public class Cexample{
      String message;
      public Cexample(String message){
            this.message = message ;
             printMessage();
      }
      public void printMessage(){
            System.out.println(message);
      }
}
```

```
public abstract class C1example{
    String message;
    public C1example(String message){
          this.message = message ;
           printMessage();      }
    public abstract String getMessage(String message);
    public void printMessage(){
          message = getMessage(message);
```



Conclusion

Code obfuscation does not provide an application with absolute protection against a malicious reverse engineering attack. Obfuscation is a cheap way of making reverse engineering so technically difficult that it becomes economically infeasible.

Finding new obfuscation techniques is a sophisticated and challenging problem.

Questions

Do you think that the more complicated

the obfuscating transformation is , the better ?