# Allatori Java Obfuscator

1. Tính năng

* Làm rối mã nguồn
* Giảm dung lượng mã nguồn
* Tăng tốc phần mềm
* Hỗ trợ Watermark
* Phần mềm miễn phí
* Khó sử dụng, cần phải hiểu các option và viết bằng tay

1. Chiến lược
2. Name Obfuscation

Là một đặc trưng của Allatori, Name Obfuscation bao gồm làm rối tên lớp (class), tên phương thức (method) và tên tham số (parameter). Kiến trúc của phần mềm không quan trọng đối với Allatori. Allatori có khả năng hiểu và làm việc với bất kỳ dự án phần mềm nào. Allatory sẽ đặt 1 tên cho nhiều biến nhất có thể, điều đó có nghĩa là 1 tên có thể đặt cho tên class, tên method và cả tên các parameter. Do đó dung lượng của phần mềm sẽ giảm xuống. Ta có ví dụ như sau :

*Original source:**Decompiler với JAD*



* Hầu như source code chỉ thay đổi rất ít và vẫn có thể hiểu được

*Decompiler với JAD sau khi sử dụng Allatori*

Ta thấy code gần như thay đổi hoàn toàn và không thể hiểu cũng như không thể chạy được sau khi decompiler

1. Flow Obfuscation

Name Obfuscation sẽ bảo vệ mã nguồn ở một mức nhất định mà có lẽ sẽ ngăn chặn được những người thiếu kinh nghiệm hoặc lười biếng. Nhưng đối với một hacker có kinh nghiệm thì sẽ không mất quá nhiều thời gian để có thể dịch ngược. Do đó, Flow Obfuscation được tạo ra nhằm mục đich thay đổi trật tự có sẵn của Java như vòng lặp, các lệnh điều kiện … Sau đây là một ví dụ hoàn chỉnh về một function Java decompiler với Allatori :

*Original source:*

Decompiler với JAD



*Decompiler với JAD sau khi sử dụng Allatori*



1. String Encryption

Bất kỳ ứng dụng nào đều chứa dữ liệu chuỗi (string). Chuỗi này có thể là text trong các phím tắt, các nút bấm, các thông báo … Các chuỗi này có thể sắp xếp các thông tin giúp hacker có thể xác định, khoanh vùng các mảng cần thiết của mã nguồn và do đó sẽ giúp hacker dễ dang hiểu các thuật toán trong chương trình của bạn. Vì vậy, mã hóa dữ liệu chuỗi là vô cùng cần thiết. Allatori sẽ tìm tất cả các chuỗi trong class và mã hóa nó.

*Original source :*

*Decompiler với JAD*

 *Decompiler với JAD sau khi sử dụng Allatori*



# PreEmptive Protection dashO for Java

## Tính năng

* Làm rối mã nguồn
* Giảm dung lượng mã nguồn
* Tăng tốc phần mềm
* Hỗ trợ Watermark
* Phần mềm tính phí
* Dễ sử dụng, có giao diện thiết lập đầy đủ
* Hỗ trợ tính năng SelfLife

## Chiến lược

* + - 1. Renaming

Renaming sẽ đổi tên các method, variable … làm mã nguồn khó hiểu. DashO sử dụng Overload Induction™ đổi tên nhiều nhất có thể các method sang cùng một tên. Ví dụ sau đây sẽ mô tả tính năng này

*Original Source :*

*Source dịch ngược sau khi sử dụng Overload Induction*



* + - 1. Control Flow
      2. String Encryption
      3. SelfLife

Tính năng này cho phép bạn gia hạn sử dụng phần mềm tới một thời điểm nhất định. Phù hợp với các phần mềm beta test

*Function solved khi decompiler với JAD sau khi sử dụng dashO*



# Java Archive Grinder

## Tính năng

* Xóa bỏ những thông tin không sử dụng (class, field, method …)
* Rút gọn tên class, field, method
* Bytecode Optimizing
* Giao diện command line

## Chiến lược

* + - 1. Removal
      2. Renaming

## Đánh giá

* Release Date : 2003/01/29 -> đã quá lâu không update
* Không còn phù hợp với phiên bản jdk hiện tại, khi test vẫn chạy, tuy nhiên output hầu như không khác input

# ProGuard

## Tính năng

* Làm rối mã nguồn
* Giảm dung lượng mã nguồn
* Tối ưu hóa mã nguồn
* Giao diện command line và gui

## Chiến lược

* + - 1. Shrinking

Proguard xác định những class và class member được sử dụng. Tất cả các class còn lại bị loại bỏ

* + - 1. Optimization

Proguard xác định các lớp và phương thức không phải là entry points có thể chuyển sang private, static hoặc final. Những parameter không sử dụng có thể bị loại bỏ.

* + - 1. obfuscation step

Proguard đổi tên các class và class member không phải là entry points và đảm bảo các class này vẫn có thể truy cập đc

## Đánh giá

* Khả năng làm rối mã nguồn chưa tốt, vẫn có thể dịch ngược và dễ dàng hiểu mã nguồn
* Dung lượng mã nguồn có giảm