Extending and Evaluating a Control Flow Obfuscation Technique for JVM Applications Utilizing `invokedynamic` with Native Bootstrapping

Bachelor's Thesis

Andre Blanke

Juli 16, 2022

Abstract

TODO

Contents

1 Motivation

2 Background

2.1 Obfuscation

2.2 Java Native Interface

2.3 `invokedynamic`

2.4 Proposed Technique

3 Implementation

3.1 Obfuscation Process

```
.class public HelloWorld
.super java/lang/Object

.method public static main : ([Ljava/lang/String;)V
    .stack  2
    .locals 1
    getstatic java/lang/System out Ljava/io/PrintStream;
    ldc "Hello, world!"
    invokevirtual java/io/PrintStream println (Ljava/lang/Object;)V
    return
.end method
```

```
.class public HelloWorld
.super java/lang/Object

.method public static main : ([Ljava/lang/String;)V
    .stack  2
    .locals 1
    invokestatic HelloWorld out ()Ljava/io/PrintStream;
    ldc "Hello, world!"
    invokevirtual java/io/PrintStream println (Ljava/lang/Object;)V
    return
.end method

.method private static synthetic out : ()Ljava/io/PrintStream;
    .stack  1
    .locals 0
    getstatic java/lang/System out Ljava/io/PrintStream;
    areturn
.end method
```

3.2 Limitations

4 Evaluation

4.1 Performance Overhead

4.2 Bytecode Size Inflation

4.3 Obfuscation Level

4.3.1 Ease of Recognition

4.3.2 Attack Resilience

5 Future Work

6 Conclusion

7 References