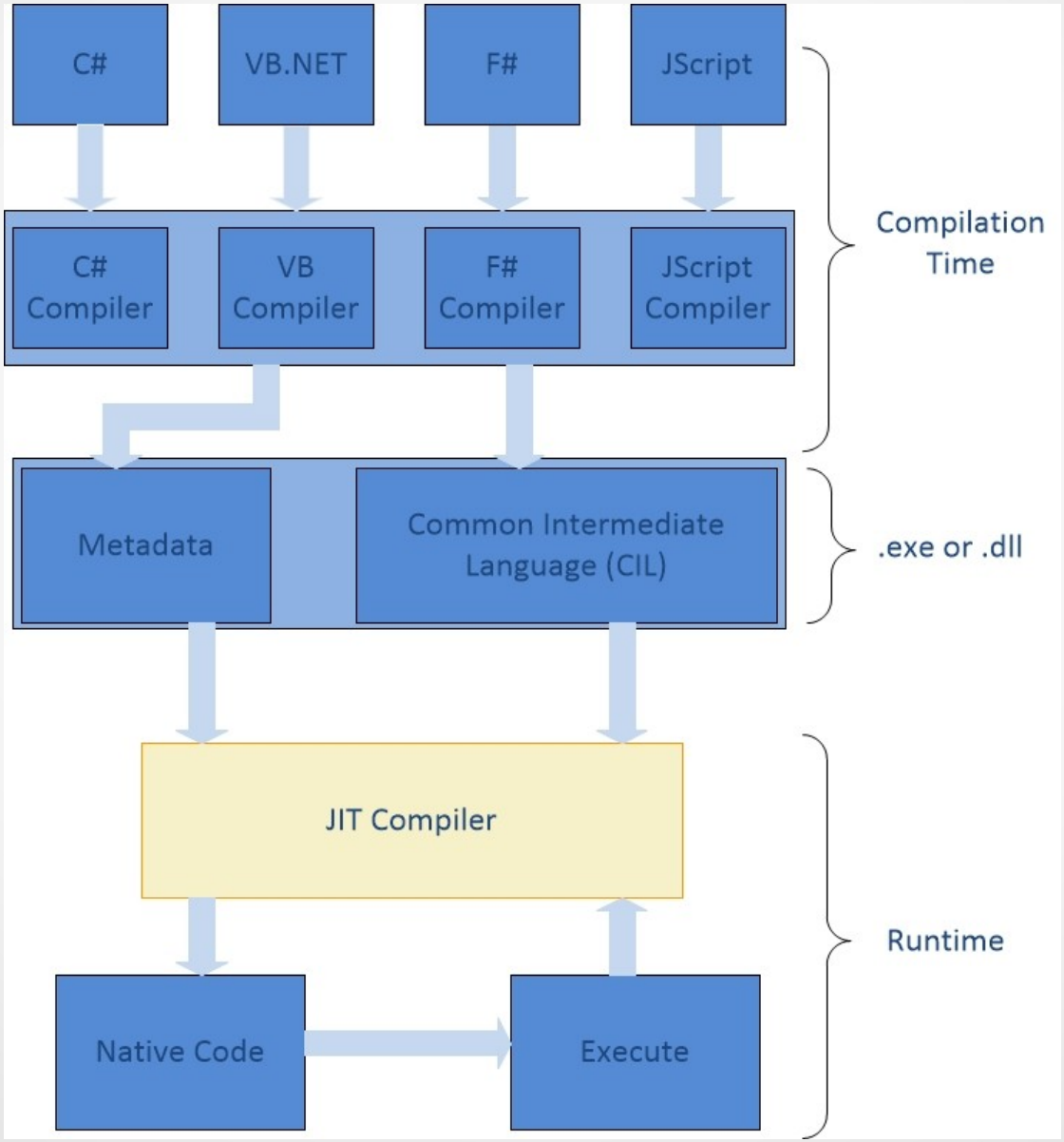
# Introduction .NET Reverse Engineering

Eric DePree

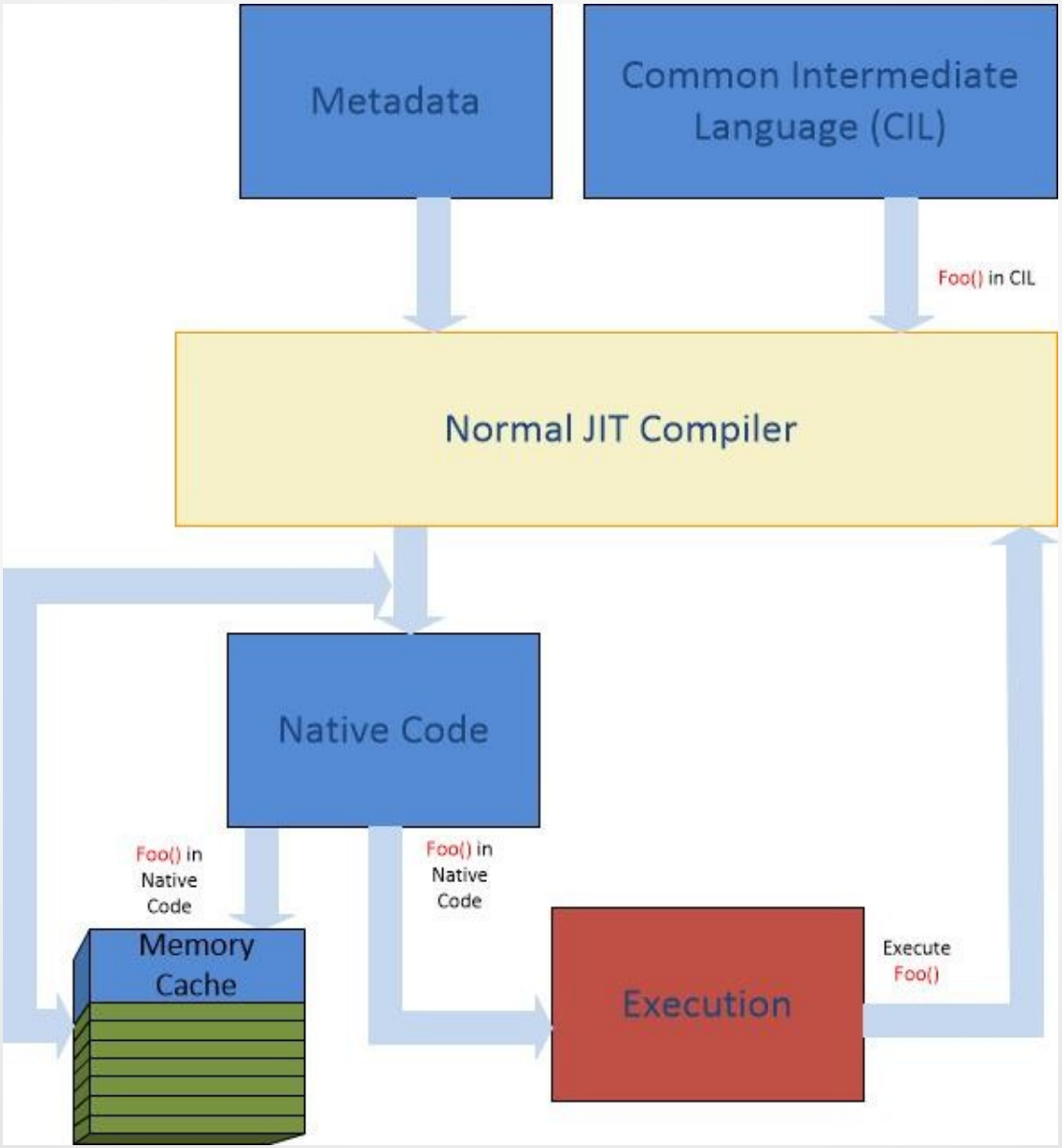
# Introduction to .NET

“Programs written for .NET are easy to reverse engineer. This is not in any way a fault in the design of .NET; it is simply a reality of modern, intermediate-compiled languages.” ~MSDN

# Common Language Runtime (CLR)



# Decompiling



# Obfuscation

- No security through obscurity!
- Make source code more difficult to reverse engineer
  - Confuse decompilers
  - Confuse humans
  - Keep logic intact

# Obfuscation

**Original Source Code Before Obfuscation**

```
private void CalcPayroll(SpecialList employeeGroup) {
   while (employeeGroup.HasMore()) {
        employee = employeeGroup.GetNext(true);
        employee.UpdateSalary();
        DistributeCheck(employee);
    }
}
```

**Reverse-Engineered Source Code**
**After Overload Induction Dotfuscation**

```
private void a(a b) {
    while (b.a()) {
        a = b.a(true);
        a.a();
        a(a);
    }
}
```

# Obfuscation

**Original Source Code Before Obfuscation**
**© 2001, Microsoft Corporation**
**(Snippet from WordCount.cs C# example code)**

```
public int CompareTo(Object o) {
    int n = occurrences - ((WordOccurrence)o).occurrences;
    if (n == 0) {
        n = String.Compare(word, ((WordOccurrence)o).word);
    }
    return(n);
}
```

**Reverse-Engineered Source Code**
**After Control Flow Obfuscation**
**By Dotfuscator Professional Edition**

```
public virtual int _a(Object A_0) {
    int local0;
    int local1;
    local0 = this.a - (c) A_0.a;
    if (local0 != 0) goto i0;
    goto i1;
    while (true) {
       return local1;
       i0: local1 = local0;
    }
    i1: local0 = System.String.Compare(this.b, (c) A_0.b);
    goto i0;
}
```

Demo

Questions? Comments?

http://edepree.com/contact

# References

- MSDN .NET Overview
- JIT Overview
- JIT Overview
- Obfuscation