

REVERSE ENGINEERING – CLASS 0x07

.NET AND JAVA

Cristian Rusu

LAST TIME

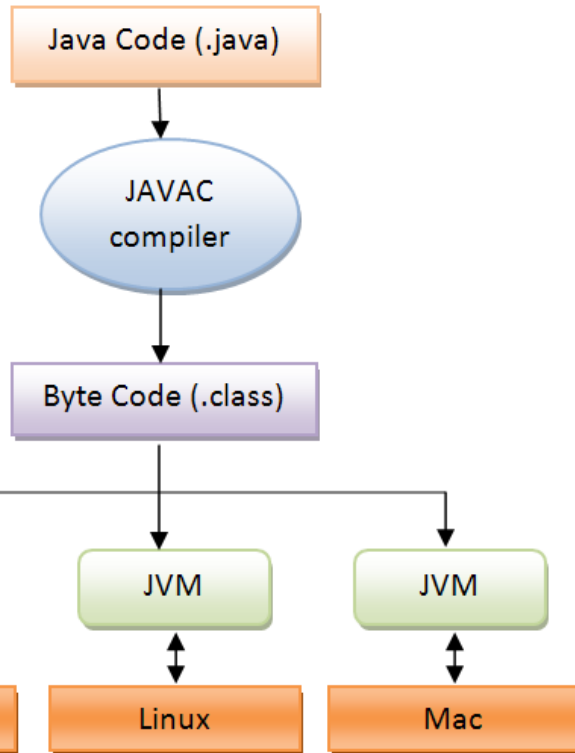
- ASLR/PIE
- RELRO
- ROP

TODAY

- Running code that is not native
- .NET RE
- Java RE

FROM SOURCE CODE TO EXECUTION

- **bytecode (non-native code):** instructions are interpreted and this interpretation goes then to the CPU (knows only machine code)



Interpreted code:

Java: java byte-code

C#: Common Intermediate Language (CIL)

Python: .py, python byte-code (fișiere .pyc)

Javascript: .js

Interpreter:

Java: Java VM

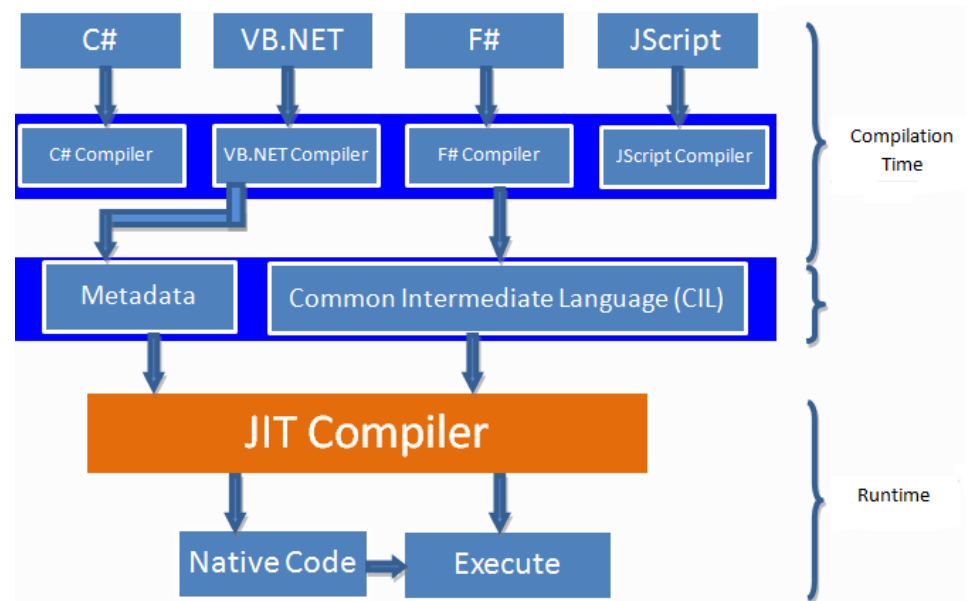
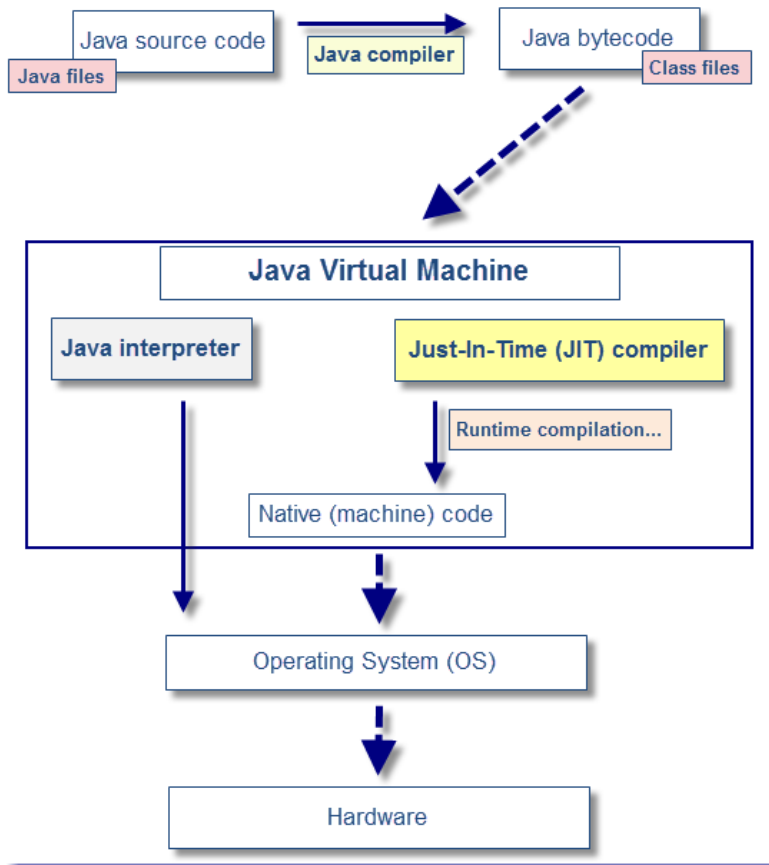
C#: Common Language Runtime (CLR) în .NET

Python: python Virtual Machine

Javascript: V8 sau Spider Monkey

FROM SOURCE CODE TO EXECUTION

- **bytecode (non-native code):** instructions are interpreted and this interpretation goes then to the CPU (knows only machine code)
- in principle, things are slower
- **JIT compilation (Just-In-Time compilation)** helps a lot



WHO CARES? (ALMOST) EVERYONE

Worldwide, Apr 2023 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	27.43 %	-0.8 %
2		Java	16.41 %	-1.7 %
3		JavaScript	9.57 %	+0.3 %
4		C#	6.9 %	-0.3 %
5		C/C++	6.65 %	-0.5 %
6		PHP	5.17 %	-0.5 %
7		R	4.22 %	-0.4 %
8		TypeScript	2.89 %	+0.5 %
9	↑	Swift	2.31 %	+0.2 %
10	↓	Objective-C	2.09 %	-0.1 %
11	↑↑↑	Rust	2.08 %	+0.9 %
12	↑	Go	1.92 %	+0.5 %
13	↓	Kotlin	1.83 %	+0.2 %
14	↓↓↓	Matlab	1.73 %	-0.2 %

BYTECODE WHICH IS COMPILED

- **bundles exist, packages that contain**
 - bytecode (intermediate language)
 - configuration
 - dependencies
 - interpreter
- **for python:**
 - py2exe
 - pyinstaller
- **if you can package it, you can unpackage it**
 - decompyle3

JAVA EXAMPLE

- the code

```
1
2 public class HelloWorld {
3
4 public static long gcd(long a, long b){
5     long factor= Math.min(a, b);
6     for(long loop= factor;loop > 1;loop--){
7         if(a % loop == 0 && b % loop == 0){
8             return loop;
9         }
10    }
11    return 1;
12 }
13
14
15 public static void main(String[] args) {
16     // Prints "Hello, World" to the terminal window.
17     System.out.println("Hello, World");
18 }
19
20 }
```


JAVA EXAMPLE

- the hexeditor view

```
00000000 ca fe ba be 00 00 00 37 00 25 0a 00 07 00 13 0a |.....7.%.....|
00000010 00 14 00 15 09 00 16 00 17 08 00 18 0a 00 19 00 |.....|
00000020 1a 07 00 1b 07 00 1c 01 00 06 3c 69 6e 69 74 3e |.....<init>|
00000030 01 00 03 28 29 56 01 00 04 43 6f 64 65 01 00 0f |...()V...Code...|
00000040 4c 69 6e 65 4e 75 6d 62 65 72 54 61 62 6c 65 01 |LineNumberTable.|
00000050 00 03 67 63 64 01 00 05 28 4a 4a 29 4a 01 00 0d |..gcd...(JJ)J...|
00000060 53 74 61 63 6b 4d 61 70 54 61 62 6c 65 01 00 04 |StackMapTable...|
00000070 6d 61 69 6e 01 00 16 28 5b 4c 6a 61 76 61 2f 6c |main...([Ljava/l|
00000080 61 6e 67 2f 53 74 72 69 6e 67 3b 29 56 01 00 0a |ang/String;)V...|
00000090 53 6f 75 72 63 65 46 69 6c 65 01 00 0f 48 65 6c |SourceFile...Hel|
000000a0 6c 6f 57 6f 72 6c 64 2e 6a 61 76 61 0c 00 08 00 |loWorld.java....|
000000b0 09 07 00 1d 0c 00 1e 00 0d 07 00 1f 0c 00 20 00 |.....|
000000c0 21 01 00 0c 48 65 6c 6c 6f 2c 20 57 6f 72 6c 64 |!...Hello, World|
000000d0 07 00 22 0c 00 23 00 24 01 00 0a 48 65 6c 6c 6f |..".#.$...Hello|
000000e0 57 6f 72 6c 64 01 00 10 6a 61 76 61 2f 6c 61 6e |World...java/lan|
000000f0 67 2f 4f 62 6a 65 63 74 01 00 0e 6a 61 76 61 2f |g/Object...java/|
00000100 6c 61 6e 67 2f 4d 61 74 68 01 00 03 6d 69 6e 01 |lang/Math...min.|
00000110 00 10 6a 61 76 61 2f 6c 61 6e 67 2f 53 79 73 74 |..java/lang/Syst|
00000120 65 6d 01 00 03 6f 75 74 01 00 15 4c 6a 61 76 61 |em...out...Ljava|
00000130 2f 69 6f 2f 50 72 69 6e 74 53 74 72 65 61 6d 3b |/io/PrintStream;|
00000140 01 00 13 6a 61 76 61 2f 69 6f 2f 50 72 69 6e 74 |...java/io/Print|
00000150 53 74 72 65 61 6d 01 00 07 70 72 69 6e 74 6c 6e |Stream...println|
00000160 01 00 15 28 4c 6a 61 76 61 2f 6c 61 6e 67 2f 53 |... (Ljava/lang/S|
00000170 74 72 69 6e 67 3b 29 56 00 21 00 06 00 07 00 00 |tring;)V.!.....|
00000180 00 00 00 03 00 01 00 08 00 09 00 01 00 0a 00 00 |.....|
00000190 00 1d 00 01 00 01 00 00 00 05 2a b7 00 01 b1 00 |.....*.....|
000001a0 00 00 01 00 0b 00 00 00 06 00 01 00 00 00 02 00 |.....|
000001b0 09 00 0c 00 0d 00 01 00 0a 00 00 00 6f 00 04 00 |.....0...|
000001c0 08 00 00 00 32 1e 20 b8 00 02 37 04 16 04 37 06 |....2. ...7...7.|
000001d0 16 06 0a 94 9e 00 21 1e 16 06 71 09 94 9a 00 0f |.....!...q.....|
000001e0 20 16 06 71 09 94 9a 00 06 16 06 ad 16 06 0a 65 |...q.....e|
000001f0 37 06 a7 ff de 0a ad 00 00 02 00 0b 00 00 00 00 |7.....|
00000200 1a 00 06 00 00 00 05 00 07 00 06 00 12 00 07 00 |.....|
00000210 24 00 08 00 27 00 06 00 30 00 0b 00 0e 00 00 00 |$....'...0.....|
00000220 0b 00 03 fd 00 0b 04 04 1b fa 00 08 00 09 00 0f |.....|
00000230 00 10 00 01 00 0a 00 00 00 25 00 02 00 01 00 00 |.....%.....|
00000240 00 09 b2 00 03 12 04 b6 00 05 b1 00 00 00 01 00 |.....|
00000250 0b 00 00 00 0a 00 02 00 00 00 11 00 08 00 12 00 |.....|
00000260 01 00 11 00 00 00 02 00 12 |.....|
00000269
```

JAVA EXAMPLE

- the reversed engineered code

```
1
2 public class HelloWorld {
3
4 public static long gcd(long a, long b){
5     long factor= Math.min(a, b);
6     for(long loop= factor;loop > 1;loop--){
7         if(a % loop == 0 && b % loop == 0){
8             return loop;
9         }
10    }
11    return 1;
12 }
13
14
15 public static void main(String[] args) {
16     // Prints "Hello, World" to the terminal window.
17     System.out.println("Hello, World");
18 }
19
20 }
```

```
public class HelloWorld
{
    public static long gcd(long paramLong1, long paramLong2) {
5        long l1 = Math.min(paramLong1, paramLong2); long l2;
6        for (l2 = l1; l2 > 1L; l2--) {
7            if (paramLong1 % l2 == 0L && paramLong2 % l2 == 0L) {
8                return l2;
9            }
10        }
11        return 1L;
12    }
13
14
15
16
17    public static void main(String[] paramArrayOfString) { System.out.println("Hello, World"); }
18 }
19 }
```

JAVA IN APK

- Android Application Package

The screenshot displays the Bytecode Viewer 2.9.22 interface. On the left, the 'Files' pane shows the directory structure of an Android application package (android.apk), including the 'com/flareon/flare' package. The 'MainActivity.class' file is selected. The 'Work Space' pane on the right shows the decompiled Java code for MainActivity.class, which extends ActionBarActivity. The code includes imports for various Android classes and defines methods for onCreate and validateEmail.

```
Bytecode Viewer 2.9.22 - https://bytecodeviewer.com | https://the.bytecode.club - @Ko

File View Settings Plugins

Files
  android.apk
  android
    support
      annotation
      v4
      v7
    com
      flareon
        flare
          BuildConfig.class
          MainActivity.class
          R$anim.class
          R$attr.class
          R$bool.class
          R$color.class
          R$dimen.class
          R$drawable.class
          R$id.class
          R$integer.class
          R$layout.class
          R$menu.class
          R$mipmap.class
          R$string.class
          R$style.class
          R$styleable.class
          R.class
          ValidateActivity.class
        Decoded Resources
        lib
        META-INF
        res
          anim
          color
          color-v11
          drawable
          drawable-hdpi-v4
          drawable-ldrtl-hdpi-v4
          drawable-ldrtl-mdpi-v4
          drawable-ldrtl-xhdpi-v4
          drawable-ldrtl-xxhdpi-v4
          drawable-ldrtl-xxxhdpi-v4

Work Space
  com/flareon/flare/MainActivity.class x
  Procyon Decompiler - Editable: true
  1 package com.flareon.flare;
  2
  3 import android.support.v7.app.*;
  4 import android.os.*;
  5 import android.view.*;
  6 import android.content.*;
  7 import android.widget.*;
  8
  9 public class MainActivity extends ActionBarActivity
  10 {
  11     public static final String EXTRA_MESSAGE = "com.flare_on.flare.MESSAGE";
  12
  13     protected void onCreate(final Bundle bundle) {
  14         super.onCreate(bundle);
  15         this setContentView(2130968601);
  16     }
  17
  18     public void validateEmail(final View view) {
  19         final Intent intent = new Intent((Context)this, (Class)ValidateActivity.class);
  20         intent.putExtra("com.flare_on.flare.MESSAGE", ((EditText)this.findViewById(213149294));
  21         this.startActivity(intent);
  22     }
  23 }
  24
```

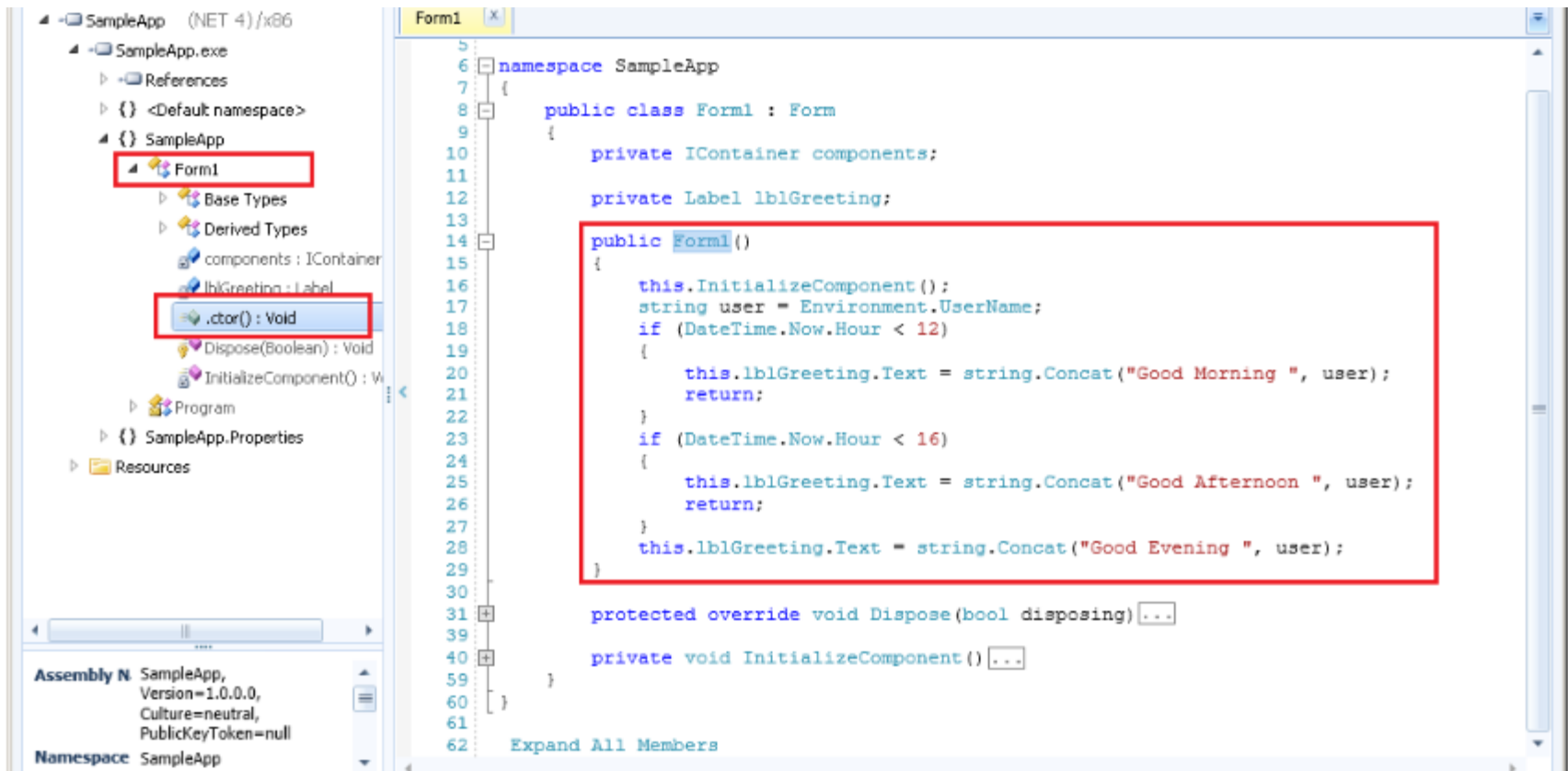
C# EXAMPLE

- the code

```
public Form1()
{
    InitializeComponent();
    string user = Environment.UserName;
    if (DateTime.Now.Hour < 12)
    {
        lblGreeting.Text = "Good Morning " + user;
    }
    else if (DateTime.Now.Hour < 16)
    {
        lblGreeting.Text = "Good Afternoon " + user;
    }
    else
    {
        lblGreeting.Text = "Good Evening " + user;
    }
}
```

C# EXAMPLE

- the reversed engineered code



The screenshot displays the Visual Studio IDE with the following components:

- Solution Explorer (Left):** Shows the project structure for 'SampleApp (NET 4)/x86'. The 'Form1' class is highlighted with a red box. Below it, the '.ctor(): Void' method is also highlighted with a red box.
- Code Editor (Right):** Shows the source code for 'Form1'. The code is as follows:

```
5
6 namespace SampleApp
7 {
8     public class Form1 : Form
9     {
10         private IContainer components;
11
12         private Label lblGreeting;
13
14         public Form1()
15         {
16             this.InitializeComponent();
17             string user = Environment.UserName;
18             if (DateTime.Now.Hour < 12)
19             {
20                 this.lblGreeting.Text = string.Concat("Good Morning ", user);
21                 return;
22             }
23             if (DateTime.Now.Hour < 16)
24             {
25                 this.lblGreeting.Text = string.Concat("Good Afternoon ", user);
26                 return;
27             }
28             this.lblGreeting.Text = string.Concat("Good Evening ", user);
29         }
30
31         protected override void Dispose(bool disposing) {...}
32
33         private void InitializeComponent() {...}
34     }
35 }
```

TOOLS TO “DECOMPILE”

- **in the lab session you will use:**
 - Bytecode Viewer
 - dnSpy
 - CFF Explorer

WHAT WE DID TODAY

- .NET RE
- Java RE

NEXT TIME ...

- **RE review**
- **anti-RE mechanisms**
- **modern RE**
- **no lab session, come for feedback or if you have questions**

REFERENCES

- Java bytecode reverse engineering, <https://resources.infosecinstitute.com/topic/java-bytecode-reverse-engineering/>
- Bytecode Obfuscation, https://owasp.org/www-community/controls/Bytecode_obfuscation
- Thwart Reverse Engineering of Your Visual Basic .NET or C# Code, <https://learn.microsoft.com/en-us/archive/msdn-magazine/2003/november/thwart-reverse-engineering-of-your-visual-basic-net-or-csharp-code>
- Java and Java Virtual Machine security vulnerabilities and their exploitation techniques, <https://www.blackhat.com/presentations/bh-asia-02/LSD/bh-asia-02-lsd-article.pdf> (and older reference, talks about the details of executing java bytecode: class loader, bytecode verifier, security manager)

