

Introduction to Reverse Engineering Server-Side Applications for Web Developers

A brief survey of tools and techniques

Agenda

- Goals of Server-side application reverse engineering
- Java
- Dotnet
- JavaScript (Briefly)
- Anti-reverse engineering techniques

New presentation, who dis?

Nick Starke

Threat Researcher at Aruba Threat Labs within the Office of the CTO at Aruba Networks / Hewlett Packard Enterprise

- Focused on firmware security, especially in networking appliances
- Board member of SecDSM (<https://secdsm.org>)
- Lives in Bondurant!
- Moved into Security from Web Development
- Blog: <https://nstarke.github.com>
- Bandcamp: <https://nstarke.bandcamp.com>

TL;DR

- For applications that compile down to byte code (JVM / CLR, primarily) there are tools that can take a compiled dll, jar, war, exe and create a near-source code quality representation of the code.
- Except for one tool, this code cannot be recompiled from the tool output.
- There are ways to modify a compiled application without source code.
- For applications written in interpreted languages (python, ruby, javascript) there is no compiled code (usually) so Reverse engineering becomes a code-review exercise
- Obfuscation is usually enough of an impediment for Reverse Engineers

Why reverse engineer server-side applications? - Security

- As an attacker, often compiled applications contain secrets like keys and passwords
- As an attacker, you might want to modify an application without the source code (wattttttt)
 - This is possible using tools like ILDASM/ILASM for .NET and Jasper/Jasmin for Java
 - However, it is not possible for the most part with the tools presented today
 - We won't cover this in any detail in this presentation :-)

Why reverse engineer server-side applications? - Dev

- Have you lost the source code? Data loss does happen :-)
- As a developer, you may need to integrate with a product that has no documentation (legacy code anyone?)
- As a developer, you may want to analyze proprietary code to understand how it works
- As a developer, it is important to understand what an attacker can do with your production binaries from a security perspective

Java

- .java files compile down to .class files
- Based on JVM bytecode for server side apps
 - The equivalent of .NET's MSIL

Java - JD-GUI

Reverse engineering tools for Java applications

- JD-GUI (<https://github.com/java-decompiler/jd-gui>)
- `brew install jd-gui` on MacOS
- Install from github releases on Linux
- Requires JDK 1.8 specifically
- Has sufficient decompiler output
- Can output all java files in a jar

JD-GUI Screenshot

SendResult.class - Java Decompiler

File Edit Navigation Search Help

websocket-api.jar

- ClientEndpointConfig.class
- CloseReason.class
- ContainerProvider.class
- DecodeException.class
- Decoder.class
- DefaultClientEndpointConfig.class
- DeploymentException.class
- EncodeException.class
 - EncodeException
- Encoder.class
- Endpoint.class
- EndpointConfig.class
- Extension.class
- HandshakeResponse.class
 - HandshakeResponse
- MessageHandler.class
- OnClose.class
- OnError.class
- OnMessage.class
- OnOpen.class
 - OnOpen
- PongMessage.class
 - PongMessage
- RemoteEndpoint.class
- SendHandler.class
- SendResult.class
 - SendResult**
 - exception : Throwable
 - ok : boolean

ClientEndpoint.class EncodeException.class HandshakeResponse.class

OnOpen.class PongMessage.class SendResult.class

```
1 @/*
2  * Licensed to the Apache Software Foundation (ASF) under one or more
3  * contributor License agreements. See the NOTICE file distributed with
4  * this work for additional information regarding copyright ownership.
5  * The ASF licenses this file to You under the Apache License, Version 2.0
6  * (the "License"); you may not use this file except in compliance with
7  * the License. You may obtain a copy of the License at
8  *
9  * http://www.apache.org/licenses/LICENSE-2.0
10 *
11 * Unless required by applicable law or agreed to in writing, software
12 * distributed under the License is distributed on an "AS IS" BASIS,
13 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 * See the License for the specific language governing permissions and
15 * limitations under the License.
16 */
17 package javax.websocket;
18
19 public final class SendResult {
20     private final Throwable exception;
21     private final boolean ok;
22
23     public SendResult(Throwable exception) {
24         this.exception = exception;
25         this.ok = (exception == null);
26     }
27
28     public SendResult() {
29         this(null);
30     }
31
32     public Throwable getException() {
33         return exception;
34     }
35
36     public boolean isOK() {
37         return ok;
38     }
39 }
```

Java - Fernflower

Fernflower is the JetBrains Java Decompiler

- Comes bundled with IntelliJ
- Can be run from the command line directly
- Has much clearer output than JD-GUI
- No UI, outputs .java files

What about other JVM Languages?

JD-GUI Scala:

```
1  import scala.Predef$;
2
3  public final class Hello$ {
4      public static final Hello$ MODULE$;
5
6      public void main(String[] args) {
7          Predef$.MODULE$.println("Hello, world");
8      }
9
10     private Hello$() {
11         MODULE$ = this;
12     }
13 }
14
```

Dotnet

- Compiles down to MSIL (Microsoft Intermediate Language)
 - The .NET equivalent of JVM Bytecode
- This runs on the .NET CLR (Common language runtime)
- Source files are .cs files which compile to exe or dll
 - DLL's more common for web apps

Dotnet - ILSpy

ILSpy - <https://github.com/icsharpcode/ILSpy>

- Open source
- Can run on Linux/MacOS/Windows
- Sufficient Output

ILSpy Screenshot

```
using Microsoft.Activities.Build.Utilities
using ...

internal static class Utilities
{
    private const string InitializeComponentMethodName = "InitializeComponent";
    ...

    internal static Activity CreateActivity(Type type, out Exception ctorException)
    {
    }

    internal static Assembly GetLocalAssembly(BuildExtensionContext context, string errorMessage)
    {
        try
        {
            string fullPath = Path.GetFullPath(context.LocalAssembly);
            return Assembly.LoadFile(fullPath);
        }
        catch (Exception ex)
        {
            if (Fx.IsFatal(ex) || ex is BadImageFormatException)
            {
                throw;
            }
            throw FxTrace.Exception.AsError(new FileLoadException(errorMessage));
        }
    }

    internal static Type[] GetTypes(Assembly assembly)
    {
        try
        {
            return assembly.GetTypes();
        }
        catch (ReflectionTypeLoadException ex)
        {
            Exception[] loaderExceptions = ex.LoaderExceptions;
            foreach (Exception ex2 in loaderExceptions)
            {
                if (ex2 is BadImageFormatException)
                {
                    throw FxTrace.Exception.AsError(ex2);
                }
            }
            throw;
        }
    }

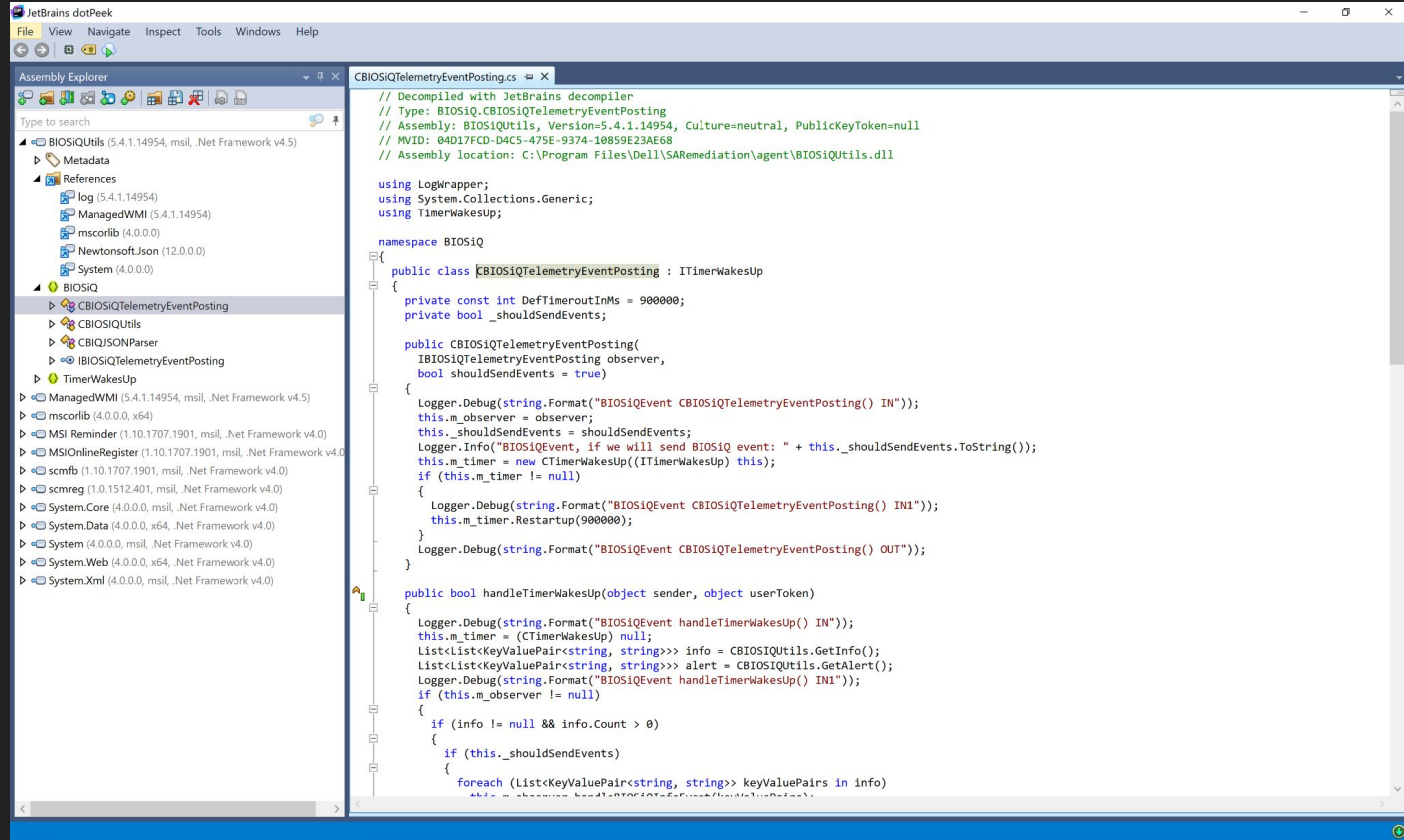
    internal static bool IsTypeAuthoredInXaml(Type type)
    {
    }
}
```

Dotnet - dotPeek

dotPeek - <https://www.jetbrains.com/decompiler/>

- JetBrains dotnet Decompiler
- Closed Source
- Free to use
- Can attempt to export DLL / EXE files as visual studio projects for recompilation

Dotpeek Screenshot



JavaScript

Not compiled down to bytecode / binary (uses JIT compilation for machine code instructions)

- Can be “minified” or “obfuscated” which makes JS difficult to read/comprehend
- Best tool to handle difficult to read JavaScript is js-beautify
- ``npm i -g js-beautify``
- Runs from CLI

Anti-reverse engineering techniques

Obfuscation!

- Dotfuscator - dotnet
- Proguard - Java

Benefits:

- Makes code extremely difficult to reverse
- Makes code extremely difficult to modify

Cons:

- Server-side: usually expensive in terms of \$ cost

Goals of Obfuscation

Obfuscation can be used to deter attackers

Usually all you need to do is put up enough of a barrier to entry that it makes a potential attacker move on to the next target

Obfuscation alone is not sufficient to secure an application!

- Secrets should not be stored in source code
- Secrets should not be stored in source code
- **SECRETS SHOULD NOT BE STORED IN SOURCE CODE**

Thank you!

Questions?

Contact:

<https://twitter.com/nstarke>

In depth presentation on dotnet / java reverse engineering coming later this fall at IADNUG and CIJUG - stay tuned!

- Blog: <https://nstarke.github.com>
- Bandcamp: <https://nstarke.bandcamp.com>