Reverse Engineering and Application Development

Iowa Code Camp 2024

whoami

Nicholas Starke

Reverse Engineer at Hewlett Packard Enterprise

Focused on firmware security

Background in Application Development

Agenda

- Why Reverse Engineering?
- Reverse Engineering Tools
- Reverse Engineering .NET
- Reverse Engineering JVM
- Reverse Engineering Native Machine Code Binaries

Why Reverse Engineering?

- To understand the internals of a given piece of software
  - Find Security Vulnerabilities
  - Analyze Malware
- To develop data integrations for proprietary software
- Modify legacy applications to fix vulnerabilities or implement new features
  - Works even when the source code or compiler toolchain is unavailable

Reverse Engineering Tools - .NET

1) dotpeek - https://www.jetbrains.com/decompiler/
2) DnSpyEx - https://github.com/dnSpyEx/dnSpy
3) ILSpy - https://github.com/icsharpcode/ILSpy
4) ildasm / ilasm

Reverse Engineering Tools - JVM

1) JD-GUI - https://java-decompiler.github.io/
2) JADX-GUI (Android) - https://github.com/skylot/jadx
3) Jasper / Jasmin - https://github.com/kohsuke/jasper / https://github.com/davidar/jasmin

Reverse Engineering Tools - Machine Code

1) Ghidra - https://ghidra-sre.org/
2) IDA Pro - https://hex-rays.com/ida-pro
3) Binary Ninja - https://binary.ninja/

Reverse Engineering - .NET

- 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.
- There are ways to modify a compiled application without source code.
    - Code signing helps mitigate the risk of this type of attack
- 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
  - This is possible using tools like ILASM.exe/ILDASM.exe for dotnet CLR

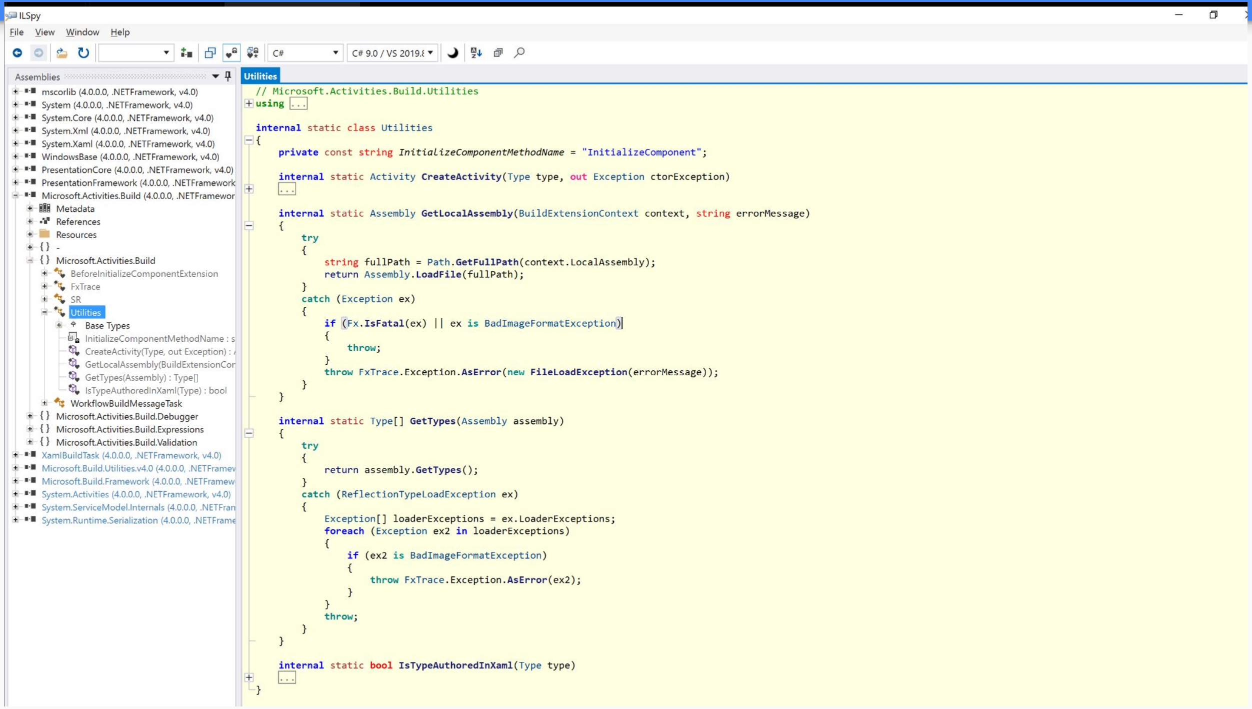
Why reverse engineer server-side applications? - Dev

- 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

.NET

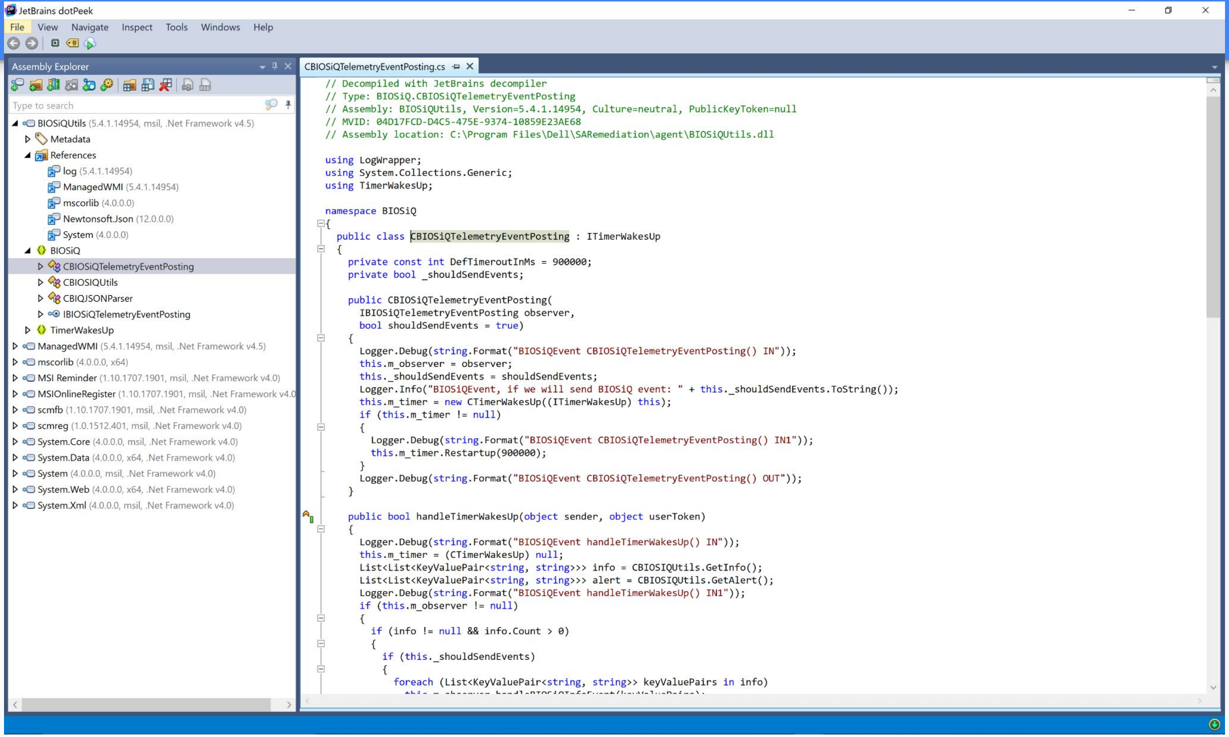
- .cs files compile down to .dll or .exe files
- Based on MSIL bytecode for server side apps
  - The equivalent of Java’s JVM Bytecode / Smali Bytecode
- Compiles down to MSIL (Microsoft Intermediate Language)
  - The .NET equivalent of JVM Bytecode
- This runs on the .NET CLR (Common language runtime)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

ILSpy


ILSpy
File View Window Help
C#
C# 9.0 / VS 2019.8
Assemblies
mscorlib (4.0.0.0, .NETFramework, v4.0)
System (4.0.0.0, .NETFramework, v4.0)
System.Core (4.0.0.0, .NETFramework, v4.0)
System.Xml (4.0.0.0, .NETFramework, v4.0)
System.Xaml (4.0.0.0, .NETFramework, v4.0)
WindowsBase (4.0.0.0, .NETFramework, v4.0)
PresentationCore (4.0.0.0, .NETFramework, v4.0)
PresentationFramework (4.0.0.0, .NETFramework
Microsoft.Activities.Build (4.0.0.0, .NETFramewor
Metadata
References
Resources
{ } -
Microsoft.Activities.Build
BeforeInitializeComponentExtension
FxTrace
SR
Utilities
Base Types
InitializeComponentMethodName : s
CreateActivity(Type, out Exception) :
GetLocalAssembly(BuildExtensionCor
GetTypes(Assembly) : Type[]
IsTypeAuthoredInXaml(Type) : bool
WorkflowBuildMessageTask
Microsoft.Activities.Build.Debugger
Microsoft.Activities.Build.Expressions
Microsoft.Activities.Build.Validation
XamlBuildTask (4.0.0.0, .NETFramework, v4.0)
Microsoft.Build.Utilities.v4.0 (4.0.0.0, .NETFramev
Microsoft.Build.Framework (4.0.0.0, .NETFramew
System.Activities (4.0.0.0, .NETFramework, v4.0)
System.ServiceModel.Internals (4.0.0.0, .NETFran
System.Runtime.Serialization (4.0.0.0, .NETFrame
Utilities

```csharp
// 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)
    ...
}
```

Dotpeek



```
BIOSiQUtils (5.4.1.14954, msil, .Net Framework v4.5)
  Metadata
  References
    log (5.4.1.14954)
    ManagedWMI (5.4.1.14954)
    mscorlib (4.0.0.0)
    Newtonsoft.Json (12.0.0.0)
    System (4.0.0.0)
  BIOSiQ
    CBIOSiQTelemetryEventPosting
    CBIOSIQUtils
    CBIQJSONParser
    IBIOSiQTelemetryEventPosting
  TimerWakesUp
ManagedWMI (5.4.1.14954, msil, .Net Framework v4.5)
mscorlib (4.0.0.0, x64)
MSI Reminder (1.10.1707.1901, msil, .Net Framework v4.0)
MSIOnlineRegister (1.10.1707.1901, msil, .Net Framework v4.0
scmfb (1.10.1707.1901, msil, .Net Framework v4.0)
scmreg (1.0.1512.401, msil, .Net Framework v4.0)
System.Core (4.0.0.0, msil, .Net Framework v4.0)
System.Data (4.0.0.0, x64, .Net Framework v4.0)
System (4.0.0.0, msil, .Net Framework v4.0)
System.Web (4.0.0.0, x64, .Net Framework v4.0)
System.Xml (4.0.0.0, msil, .Net Framework v4.0)
```

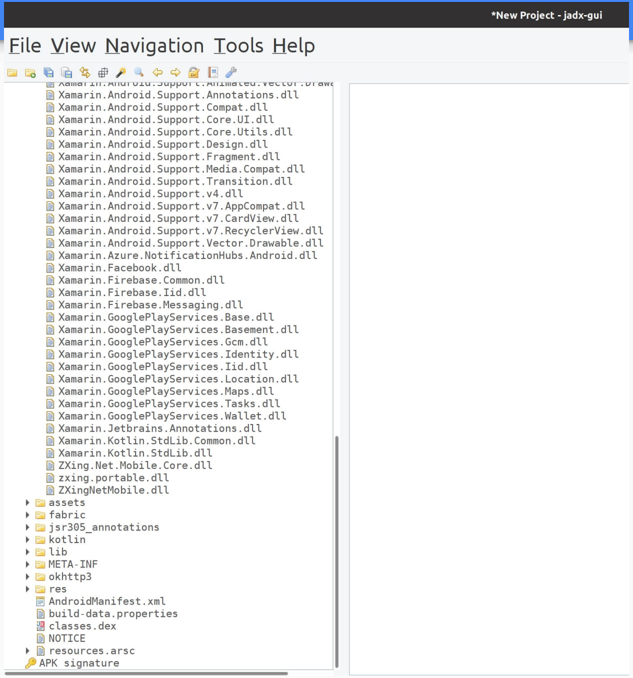
```csharp
// Decompiled with JetBrains decompiler
// Type: BIOSiQ.CBIOSiQTelemetryEventPosting
// Assembly: BIOSiQUtils, Version=5.4.1.14954, Culture=neutral, PublicKeyToken=null
// MVID: 04D17FCD-D4C5-475E-9374-10859E23AE68
// Assembly location: C:\Program Files\Dell\SARemediation\agent\BIOSiQUtils.dll

using LogWrapper;
using System.Collections.Generic;
using TimerWakesUp;

namespace BIOSiQ
{
  public class CBIOSiQTelemetryEventPosting : ITimerWakesUp
  {
    private const int DefTimeroutInMs = 900000;
    private bool _shouldSendEvents;

    public CBIOSiQTelemetryEventPosting(
      IBIOSiQTelemetryEventPosting observer,
      bool shouldSendEvents = true)
    {
      Logger.Debug(string.Format("BIOSiQEvent CBIOSiQTelemetryEventPosting() IN"));
      this.m_observer = observer;
      this._shouldSendEvents = shouldSendEvents;
      Logger.Info("BIOSiQEvent, if we will send BIOSiQ event: " + this._shouldSendEvents.ToString());
      this.m_timer = new CTimerWakesUp((ITimerWakesUp) this);
      if (this.m_timer != null)
      {
        Logger.Debug(string.Format("BIOSiQEvent CBIOSiQTelemetryEventPosting() IN1"));
        this.m_timer.Restartup(900000);
      }
      Logger.Debug(string.Format("BIOSiQEvent CBIOSiQTelemetryEventPosting() OUT"));
    }

    public bool handleTimerWakesUp(object sender, object userToken)
    {
      Logger.Debug(string.Format("BIOSiQEvent handleTimerWakesUp() IN"));
      this.m_timer = (CTimerWakesUp) null;
      List<List<KeyValuePair<string, string>>> info = CBIOSIQUtils.GetInfo();
      List<List<KeyValuePair<string, string>>> alert = CBIOSIQUtils.GetAlert();
      Logger.Debug(string.Format("BIOSiQEvent handleTimerWakesUp() IN1"));
      if (this.m_observer != null)
      {
        if (info != null && info.Count > 0)
        {
          if (this._shouldSendEvents)
          {
            foreach (List<KeyValuePair<string, string>> keyValuePairs in info)
```

Client-side .NET

Jadx-gui - https://github.com/skylot/jadx

- Useful for extracting Xamarin Assemblies
- Extracts the static content (res/) from the APK and presents it in a tree view

Client-side .NET



```
Xamarin.Android.Support.Animated.Vector.Drawa
Xamarin.Android.Support.Annotations.dll
Xamarin.Android.Support.Compat.dll
Xamarin.Android.Support.Core.UI.dll
Xamarin.Android.Support.Core.Utils.dll
Xamarin.Android.Support.Design.dll
Xamarin.Android.Support.Fragment.dll
Xamarin.Android.Support.Media.Compat.dll
Xamarin.Android.Support.Transition.dll
Xamarin.Android.Support.v4.dll
Xamarin.Android.Support.v7.AppCompat.dll
Xamarin.Android.Support.v7.CardView.dll
Xamarin.Android.Support.v7.RecyclerView.dll
Xamarin.Android.Support.Vector.Drawable.dll
Xamarin.Azure.NotificationHubs.Android.dll
Xamarin.Facebook.dll
Xamarin.Firebase.Common.dll
Xamarin.Firebase.Iid.dll
Xamarin.Firebase.Messaging.dll
Xamarin.GooglePlayServices.Base.dll
Xamarin.GooglePlayServices.Basement.dll
Xamarin.GooglePlayServices.Gcm.dll
Xamarin.GooglePlayServices.Identity.dll
Xamarin.GooglePlayServices.Iid.dll
Xamarin.GooglePlayServices.Location.dll
Xamarin.GooglePlayServices.Maps.dll
Xamarin.GooglePlayServices.Tasks.dll
Xamarin.GooglePlayServices.Wallet.dll
Xamarin.Jetbrains.Annotations.dll
Xamarin.Kotlin.StdLib.Common.dll
Xamarin.Kotlin.StdLib.dll
ZXing.Net.Mobile.Core.dll
zxing.portable.dll
ZXingNetMobile.dll
assets
fabric
jsr305_annotations
kotlin
lib
META-INF
okhttp3
res
AndroidManifest.xml
build-data.properties
classes.dex
NOTICE
resources.arsc
APK signature
```

Modifying MSIL Bytecode without Source Code

The next few slides will focus on techniques for modifying MSIL Bytecode without access to the original source code.

We'll discuss:

- Why would anyone want to do this?
- Examples
- Process
- Tooling

Why would anyone want to do this?

Development:

- Modify an application when source code is lost

Security:

- Patch an application to log out sensitive information

Examples of Patching MSIL Bytecode - Security

Server-side

- Server side: when a login request is received, log out the username and password to a file on the filesystem.

Client-side

- Client side: make an HTTP request to an unauthorized remote server with authentication tokens received from a legitimate authentication request

Process

1) Write out Dotnet code you wish to inject in a console application. Create a function that accepts the data you wish to operate on.
2) Disassemble this console application
3) Disassemble the source code you wish to inject code into
4) Modify the source code disassembly to include the console application disassembly and write integration disassembly to call the function you wrote in 1)
5) Reassemble source .cs file
6) Reassemble DLL / Drop on file system cache.

https://starkeblog.com/backdooring/dotnet/2024/04/19/backdooring-dotnet-applications.html

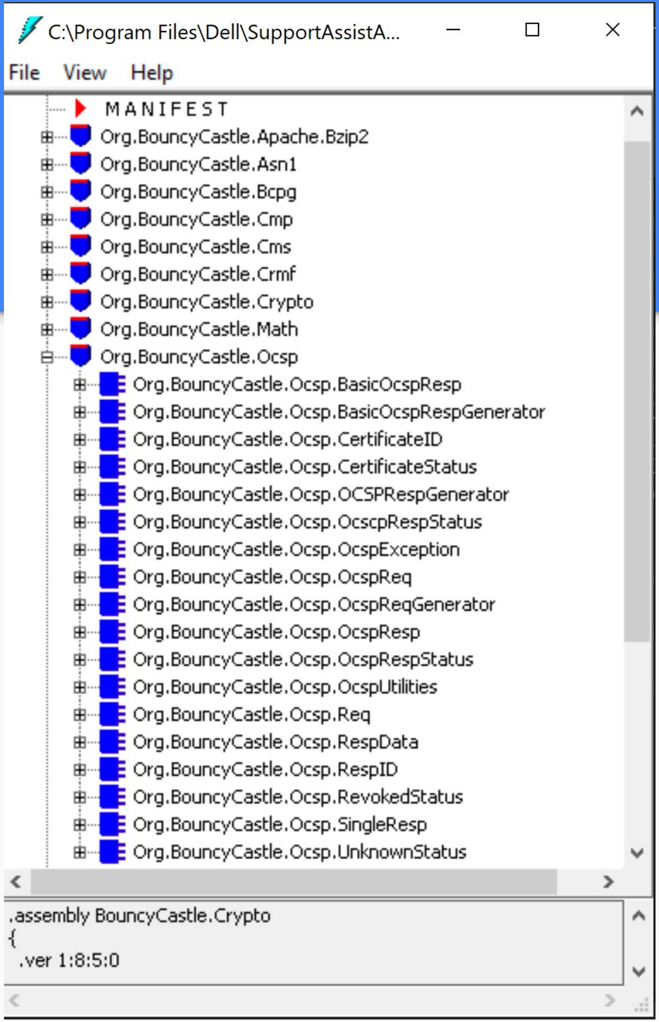
Dotnet Disassembler / Assembler Duo

Dotnet MSIL Assembler: ILASM.exe - https://docs.microsoft.com/en-us/dotnet/framework/tools/ilasm-exe-il-assembler

Dotnet MSIL Disassembler: ILDASM.exe - https://docs.microsoft.com/en-us/dotnet/framework/tools/ildasm-exe-il-disassembler

- These two tools are built to work with each other.
- Available through Visual Studio Developer Shell

ildasm

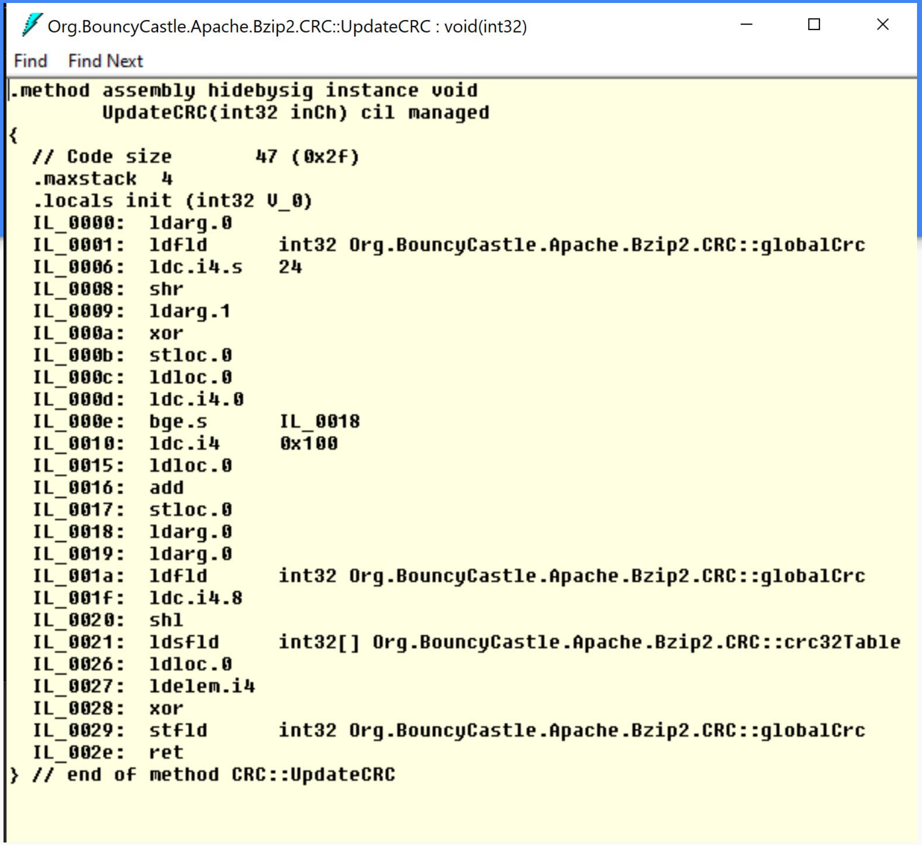

C:\Program Files\Dell\SupportAssistA...

File View Help

```
MANIFEST
⊞ Org.BouncyCastle.Apache.Bzip2
⊞ Org.BouncyCastle.Asn1
⊞ Org.BouncyCastle.Bcpg
⊞ Org.BouncyCastle.Cmp
⊞ Org.BouncyCastle.Cms
⊞ Org.BouncyCastle.Crmf
⊞ Org.BouncyCastle.Crypto
⊞ Org.BouncyCastle.Math
⊟ Org.BouncyCastle.Ocsp
    ⊞ Org.BouncyCastle.Ocsp.BasicOcspResp
    ⊞ Org.BouncyCastle.Ocsp.BasicOcspRespGenerator
    ⊞ Org.BouncyCastle.Ocsp.CertificateID
    ⊞ Org.BouncyCastle.Ocsp.CertificateStatus
    ⊞ Org.BouncyCastle.Ocsp.OCSPRespGenerator
    ⊞ Org.BouncyCastle.Ocsp.OcscpRespStatus
    ⊞ Org.BouncyCastle.Ocsp.OcspException
    ⊞ Org.BouncyCastle.Ocsp.OcspReq
    ⊞ Org.BouncyCastle.Ocsp.OcspReqGenerator
    ⊞ Org.BouncyCastle.Ocsp.OcspResp
    ⊞ Org.BouncyCastle.Ocsp.OcspRespStatus
    ⊞ Org.BouncyCastle.Ocsp.OcspUtilities
    ⊞ Org.BouncyCastle.Ocsp.Req
    ⊞ Org.BouncyCastle.Ocsp.RespData
    ⊞ Org.BouncyCastle.Ocsp.RespID
    ⊞ Org.BouncyCastle.Ocsp.RevokedStatus
    ⊞ Org.BouncyCastle.Ocsp.SingleResp
    ⊞ Org.BouncyCastle.Ocsp.UnknownStatus
```

```
.assembly BouncyCastle.Crypto
{
 .ver 1:8:5:0
```

Ildasm - Screenshot



```
.method assembly hidebysig instance void
        UpdateCRC(int32 inCh) cil managed
{
  // Code size       47 (0x2f)
  .maxstack  4
  .locals init (int32 V_0)
  IL_0000:  ldarg.0
  IL_0001:  ldfld      int32 Org.BouncyCastle.Apache.Bzip2.CRC::globalCrc
  IL_0006:  ldc.i4.s   24
  IL_0008:  shr
  IL_0009:  ldarg.1
  IL_000a:  xor
  IL_000b:  stloc.0
  IL_000c:  ldloc.0
  IL_000d:  ldc.i4.0
  IL_000e:  bge.s      IL_0018
  IL_0010:  ldc.i4     0x100
  IL_0015:  ldloc.0
  IL_0016:  add
  IL_0017:  stloc.0
  IL_0018:  ldarg.0
  IL_0019:  ldarg.0
  IL_001a:  ldfld      int32 Org.BouncyCastle.Apache.Bzip2.CRC::globalCrc
  IL_001f:  ldc.i4.8
  IL_0020:  shl
  IL_0021:  ldsfld     int32[] Org.BouncyCastle.Apache.Bzip2.CRC::crc32Table
  IL_0026:  ldloc.0
  IL_0027:  ldelem.i4
  IL_0028:  xor
  IL_0029:  stfld      int32 Org.BouncyCastle.Apache.Bzip2.CRC::globalCrc
  IL_002e:  ret
} // end of method CRC::UpdateCRC
```

How to mitigate this threat

- Strong name signing - https://docs.microsoft.com/en-us/dotnet/standard/assembly/sign-strong-name
- Read only file system for executable code

Android implements code signing by default - consider this for your production applications even when they are server-side.

Anti-reverse engineering techniques

Obfuscation!

- Dotfuscator - https://www.preemptive.com/products/dotfuscator

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

Resources

- Managed Code Rootkits (Book): https://www.amazon.com/Managed-Code-Rootkits-Hooking-Environments/dp/1597495743

Reverse Engineering Java / JVM

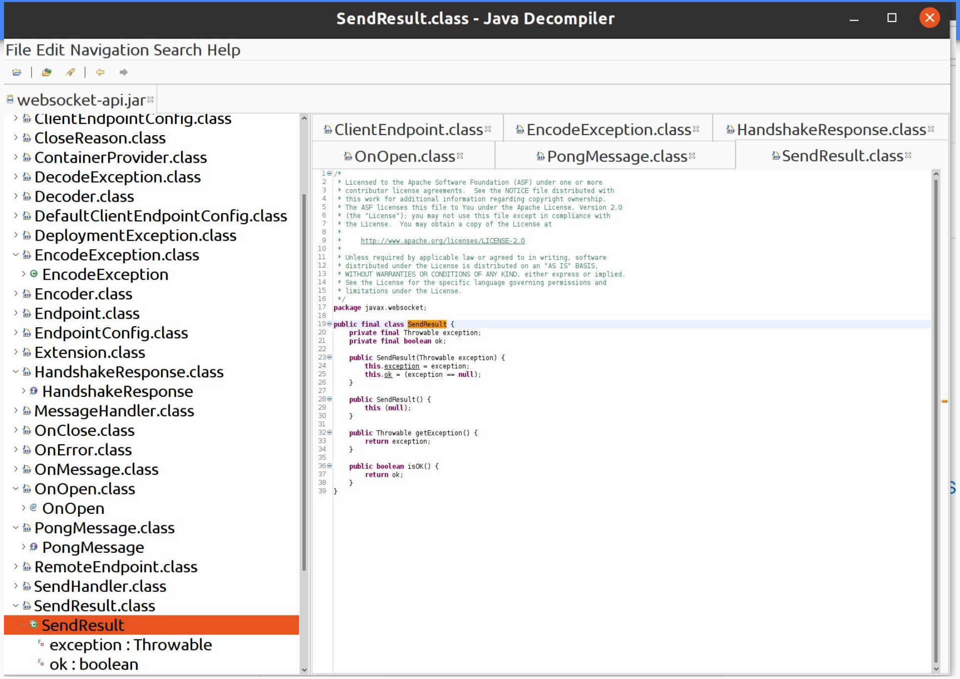
- .java files compile down to .class files
- Based on JVM bytecode for server side apps
  - The equivalent of .NET’s MSIL

Server-side Java - JD-GUI

Reverse engineering tools for Server-side 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



```
/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements.  See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License.  You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package javax.websocket;

public final class SendResult {
    private final Throwable exception;
    private final boolean ok;

    public SendResult(Throwable exception) {
        this.exception = exception;
        this.ok = (exception == null);
    }

    public SendResult() {
        this (null);
    }

    public Throwable getException() {
        return exception;
    }

    public boolean isOK() {
        return ok;
    }
}
```

Java - Fernflower Decompiler

Fernflower is the JetBrains Java Decompiler

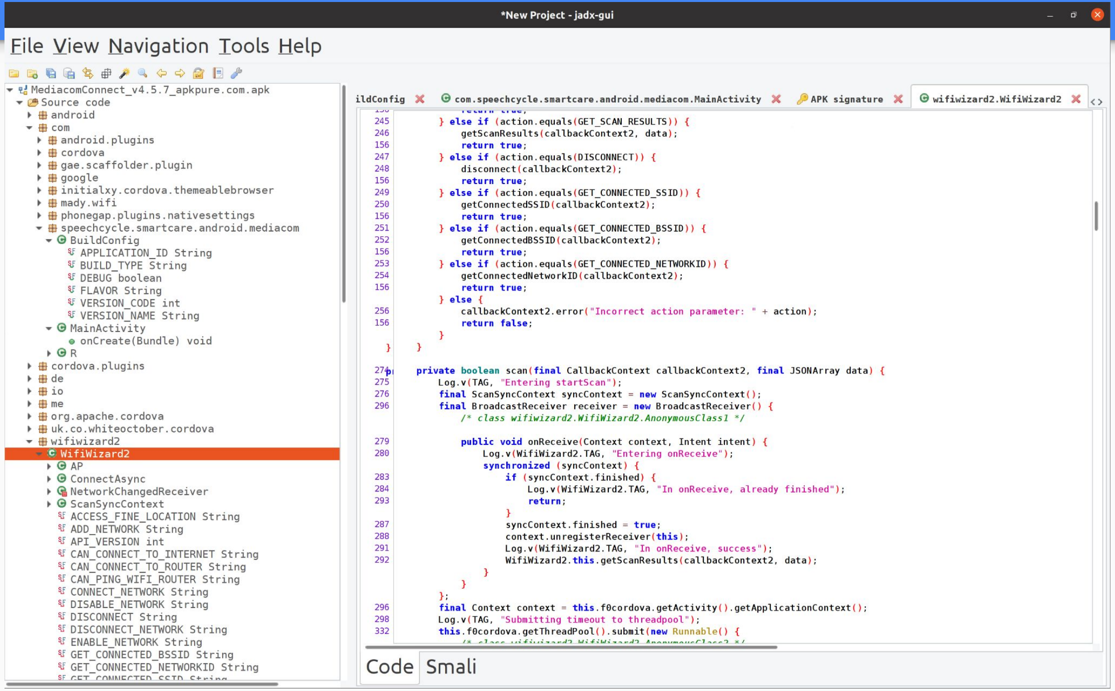
- Comes bundled with IntelliJ
- Can be run from the command line directly
- Has much clearer output than JD-GUI
- No UI, outputs .java files

Client-side Java (Native Android)

Jadx-gui - https://github.com/skylot/jadx

- A lot like JD-GUI
- Does all the manual work of extracting the APK then disassembling/decompiling the SMALI bytecode into Java classes
- Extracts the static content (res/) from the APK and presents it in a tree view

JADX-GUI Screenshot



*New Project - jadx-gui

File View Navigation Tools Help

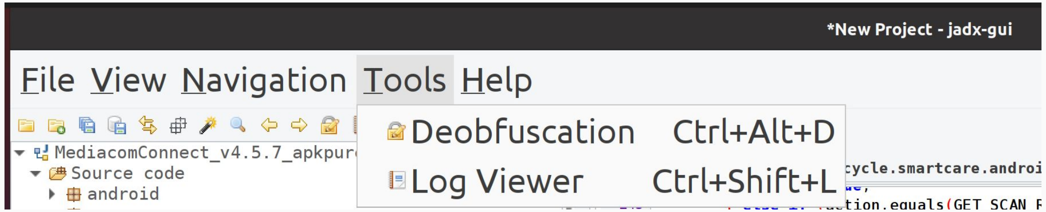
```
MediacomConnect_v4.5.7_apkpure.com.apk
  Source code
    android
    com
      android.plugins
      cordova
      gae.scaffolder.plugin
      google
      initialxy.cordova.themeablebrowser
      mady.wifi
      phonegap.plugins.nativesettings
      speechcycle.smartcare.android.mediacom
        BuildConfig
          APPLICATION_ID String
          BUILD_TYPE String
          DEBUG boolean
          FLAVOR String
          VERSION_CODE int
          VERSION_NAME String
        MainActivity
          onCreate(Bundle) void
        R
    cordova.plugins
    de
    io
    me
    org.apache.cordova
    uk.co.whiteoctober.cordova
    wifiwizard2
      WifiWizard2
        AP
        ConnectAsync
        NetworkChangedReceiver
        ScanSyncContext
        ACCESS_FINE_LOCATION String
        ADD_NETWORK String
        API_VERSION int
        CAN_CONNECT_TO_INTERNET String
        CAN_CONNECT_TO_ROUTER String
        CAN_PING_WIFI_ROUTER String
        CONNECT_NETWORK String
        DISABLE_NETWORK String
        DISCONNECT String
        DISCONNECT_NETWORK String
        ENABLE_NETWORK String
        GET_CONNECTED_BSSID String
        GET_CONNECTED_NETWORKID String
```

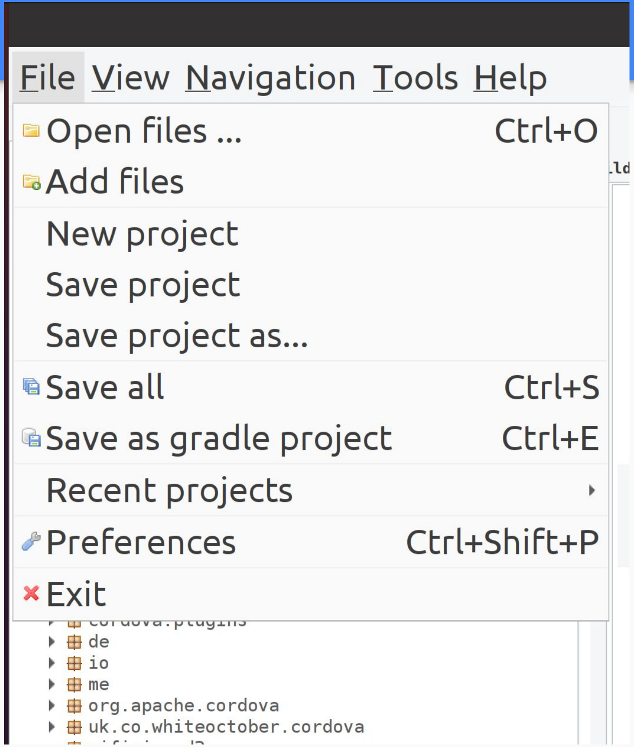
ildConfig | com.speechcycle.smartcare.android.mediacom.MainActivity | APK signature | wifiwizard2.WifiWizard2

```java
        } else if (action.equals(GET_SCAN_RESULTS)) {
            getScanResults(callbackContext2, data);
            return true;
        } else if (action.equals(DISCONNECT)) {
            disconnect(callbackContext2);
            return true;
        } else if (action.equals(GET_CONNECTED_SSID)) {
            getConnectedSSID(callbackContext2);
            return true;
        } else if (action.equals(GET_CONNECTED_BSSID)) {
            getConnectedBSSID(callbackContext2);
            return true;
        } else if (action.equals(GET_CONNECTED_NETWORKID)) {
            getConnectedNetworkID(callbackContext2);
            return true;
        } else {
            callbackContext2.error("Incorrect action parameter: " + action);
            return false;
        }
    }

    private boolean scan(final CallbackContext callbackContext2, final JSONArray data) {
        Log.v(TAG, "Entering startScan");
        final ScanSyncContext syncContext = new ScanSyncContext();
        final BroadcastReceiver receiver = new BroadcastReceiver() {
            /* class wifiwizard2.WifiWizard2.AnonymousClass1 */

            public void onReceive(Context context, Intent intent) {
                Log.v(WifiWizard2.TAG, "Entering onReceive");
                synchronized (syncContext) {
                    if (syncContext.finished) {
                        Log.v(WifiWizard2.TAG, "In onReceive, already finished");
                        return;
                    }
                    syncContext.finished = true;
                    context.unregisterReceiver(this);
                    Log.v(WifiWizard2.TAG, "In onReceive, success");
                    WifiWizard2.this.getScanResults(callbackContext2, data);
                }
            }
        };
        final Context context = this.f0cordova.getActivity().getApplicationContext();
        Log.v(TAG, "Submitting timeout to threadpool");
        this.f0cordova.getThreadPool().submit(new Runnable() {
```

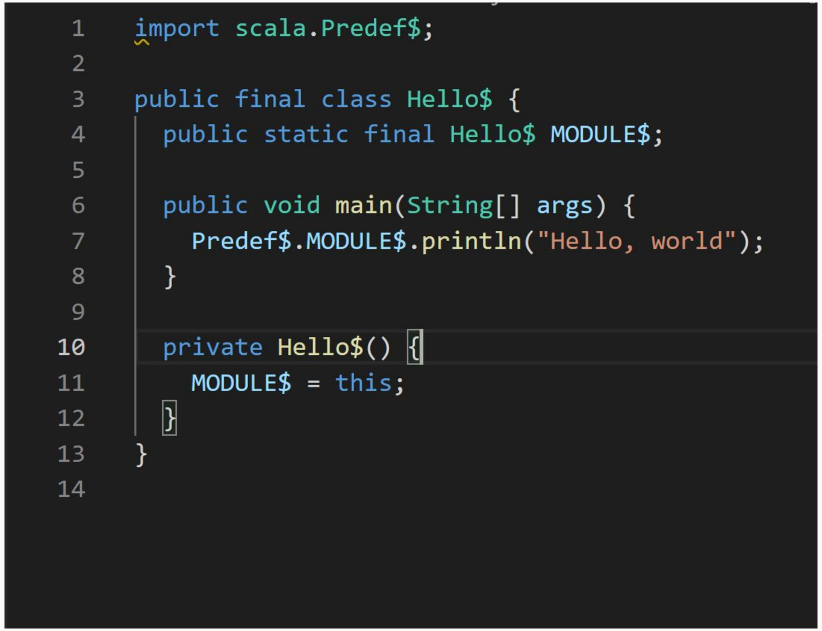
Code | Smali

More JADX-GUI Screenshots





JD-GUI - Scala Decompiler Output



```
import scala.Predef$;

public final class Hello$ {
  public static final Hello$ MODULE$;

  public void main(String[] args) {
    Predef$.MODULE$.println("Hello, world");
  }

  private Hello$() {
    MODULE$ = this;
  }
}
```

Java Disassembler / Assembler Duo

Java Class Disassembler: Jasper - https://github.com/kohsuke/jasper

Java Class Assembler: Jasmin - https://github.com/davidar/jasmin

- These two tools are built to work with each other.
- Jasmin will not work with “javap -c”!
- Both tools were built between 2000-2004
- Modifications to source are necessary for both to compile with modern Java tooling.
- Jasper works with maven, Jasmin works with ant.

How to mitigate this threat

- Jar signing (via `jarsigner` tool)
- Read only file system for executable code

Android implements jar signing by default - consider this for your production applications even when they are server-side.

JVM: Anti-reverse engineering techniques

Obfuscation!

- Proguard - Java

Benefits:

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

Cons:

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

JVM: More Resources

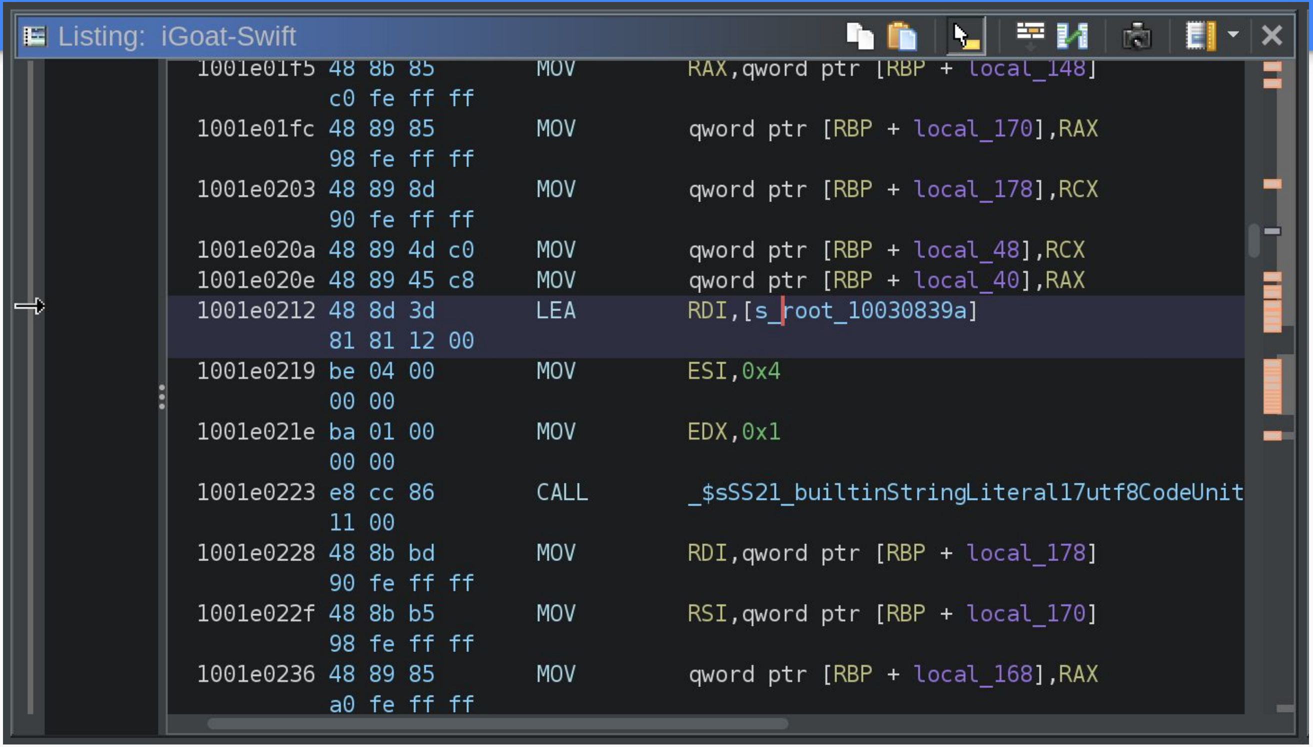
- Covert Java (Book): https://www.amazon.com/dp/0672326388

Native Machine Code

Ghidra - https://ghidra-sre.org/

- Developed by the US National Security Agency
- Open-source

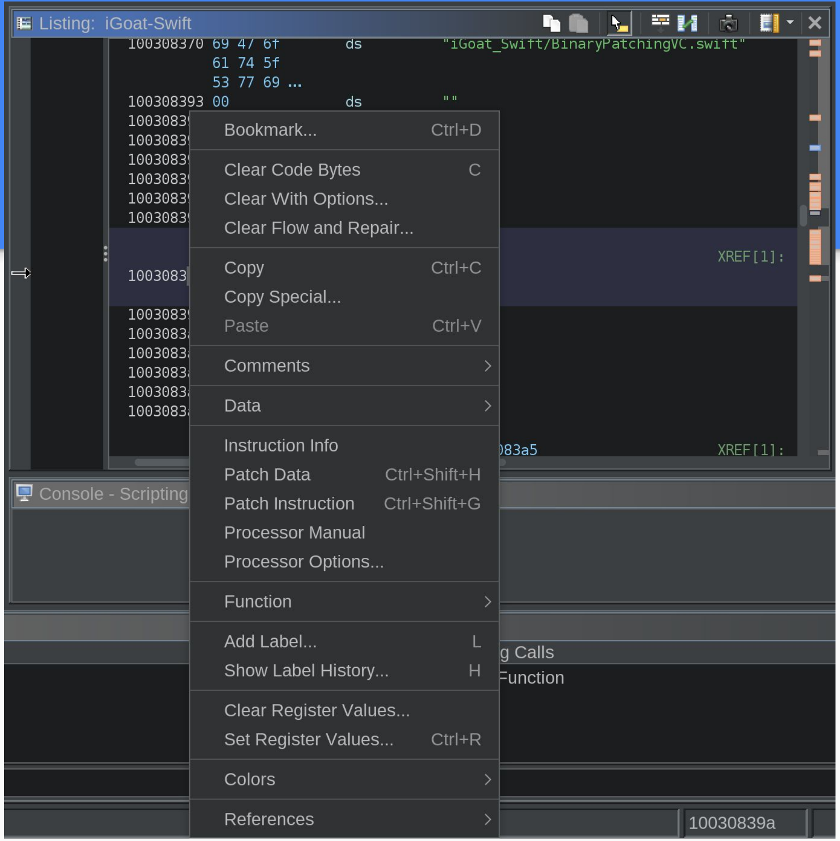
Ghidra - Disassembler



Listing: iGoat-Swift

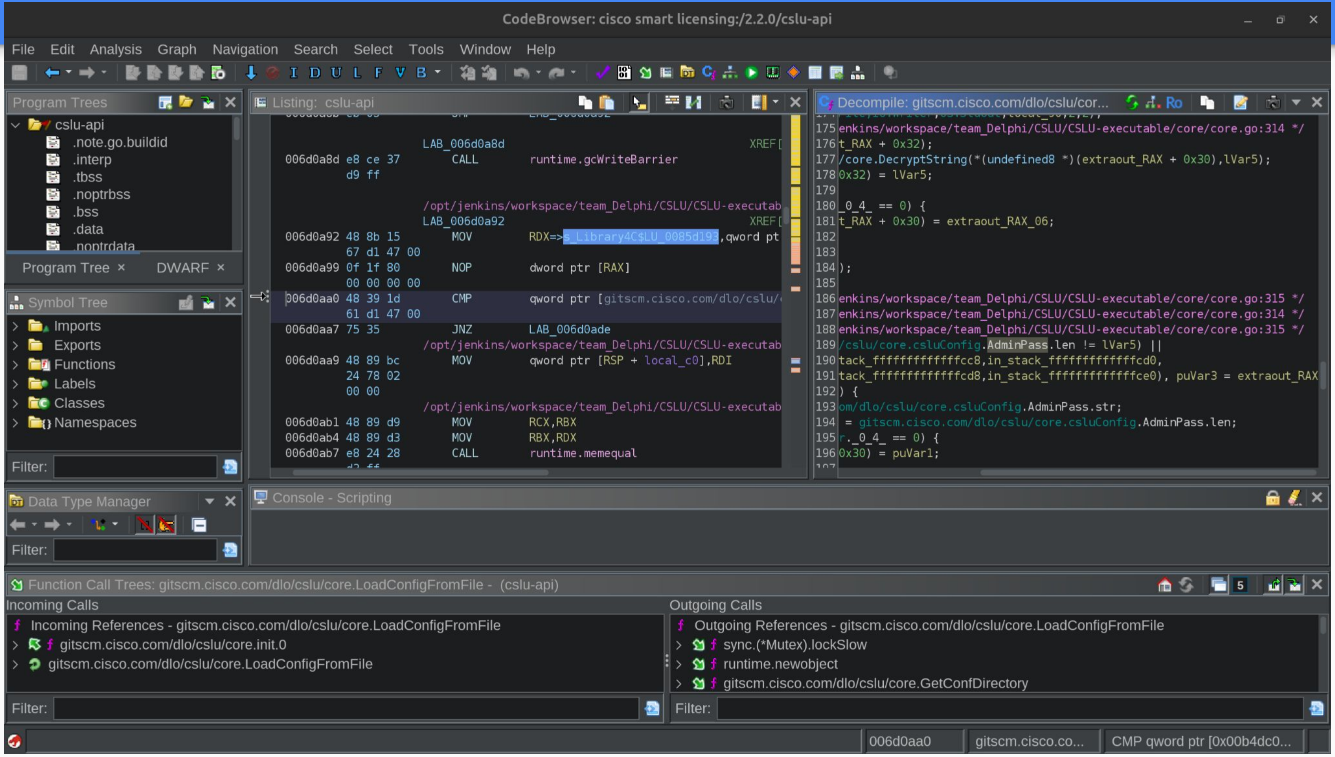
```
1001e01f5 48 8b 85          MOV      RAX,qword ptr [RBP + local_148]
          c0 fe ff ff
1001e01fc 48 89 85          MOV      qword ptr [RBP + local_170],RAX
          98 fe ff ff
1001e0203 48 89 8d          MOV      qword ptr [RBP + local_178],RCX
          90 fe ff ff
1001e020a 48 89 4d c0       MOV      qword ptr [RBP + local_48],RCX
1001e020e 48 89 45 c8       MOV      qword ptr [RBP + local_40],RAX
1001e0212 48 8d 3d          LEA      RDI,[s_root_10030839a]
          81 81 12 00
1001e0219 be 04 00          MOV      ESI,0x4
          00 00
1001e021e ba 01 00          MOV      EDX,0x1
          00 00
1001e0223 e8 cc 86          CALL     _$sSS21_builtinStringLiteral17utf8CodeUnit
          11 00
1001e0228 48 8b bd          MOV      RDI,qword ptr [RBP + local_178]
          90 fe ff ff
1001e022f 48 8b b5          MOV      RSI,qword ptr [RBP + local_170]
          98 fe ff ff
1001e0236 48 89 85          MOV      qword ptr [RBP + local_168],RAX
          a0 fe ff ff
```

Ghidra - Patch Binary



```
100308370 69 47 6f        ds        "iGoat_Swift/BinaryPatchingVC.swift"
          61 74 5f
          53 77 69 ...
100308393 00              ds        ""
```

Ghidra - Full Screenshot



CodeBrowser: cisco smart licensing:/2.2.0/cslu-api

Program Trees
- cslu-api
  - .note.go.buildid
  - .interp
  - .tbss
  - .noptrbss
  - .bss
  - .data
  - .noptrdata

Symbol Tree
- Imports
- Exports
- Functions
- Labels
- Classes
- Namespaces

Listing: cslu-api

```
                          LAB_006d0a8d                              XREF[
006d0a8d e8 ce 37         CALL       runtime.gcWriteBarrier
         d9 ff

                          /opt/jenkins/workspace/team_Delphi/CSLU/CSLU-executab
                          LAB_006d0a92                              XREF[
006d0a92 48 8b 15         MOV        RDX=>s_Library4C$LU_0085d193,qword pt
         67 d1 47 00
006d0a99 0f 1f 80         NOP        dword ptr [RAX]
         00 00 00 00
006d0aa0 48 39 1d         CMP        qword ptr [gitscm.cisco.com/dlo/cslu/
         61 d1 47 00
006d0aa7 75 35            JNZ        LAB_006d0ade
                          /opt/jenkins/workspace/team_Delphi/CSLU/CSLU-executab
006d0aa9 48 89 bc         MOV        qword ptr [RSP + local_c0],RDI
         24 78 02
         00 00
                          /opt/jenkins/workspace/team_Delphi/CSLU/CSLU-executab
006d0ab1 48 89 d9         MOV        RCX,RBX
006d0ab4 48 89 d3         MOV        RBX,RDX
006d0ab7 e8 24 28         CALL       runtime.memequal
```

Decompile: gitscm.cisco.com/dlo/cslu/cor...

```
enkins/workspace/team_Delphi/CSLU/CSLU-executable/core/core.go:314 */
t_RAX + 0x32);
/core.DecryptString(*(undefined8 *)(extraout_RAX + 0x30),lVar5);
0x32) = lVar5;

_0_4_ == 0) {
t_RAX + 0x30) = extraout_RAX_06;


);

enkins/workspace/team_Delphi/CSLU/CSLU-executable/core/core.go:315 */
enkins/workspace/team_Delphi/CSLU/CSLU-executable/core/core.go:314 */
enkins/workspace/team_Delphi/CSLU/CSLU-executable/core/core.go:315 */
/cslu/core.csluConfig.AdminPass.len != lVar5) ||
tack_fffffffffffcc8,in_stack_fffffffffffcd0,
tack_fffffffffffcd8,in_stack_fffffffffffce0), puVar3 = extraout_RAX
) {
om/dlo/cslu/core.csluConfig.AdminPass.str;
 = gitscm.cisco.com/dlo/cslu/core.csluConfig.AdminPass.len;
r._0_4_ == 0) {
0x30) = puVar1;
```

Console - Scripting

Function Call Trees: gitscm.cisco.com/dlo/cslu/core.LoadConfigFromFile - (cslu-api)

Incoming Calls
- Incoming References - gitscm.cisco.com/dlo/cslu/core.LoadConfigFromFile
  - gitscm.cisco.com/dlo/cslu/core.init.0
  - gitscm.cisco.com/dlo/cslu/core.LoadConfigFromFile

Outgoing Calls
- Outgoing References - gitscm.cisco.com/dlo/cslu/core.LoadConfigFromFile
  - sync.(*Mutex).lockSlow
  - runtime.newobject
  - gitscm.cisco.com/dlo/cslu/core.GetConfDirectory

006d0aa0 | gitscm.cisco.co... | CMP qword ptr [0x00b4dc0...

Questions?

Thank you!