Java Bayt Kod Hata Ayıklaması

written by Mert SARICA | 1 July 2015

Tarık Y. sağolsun 10 Ekim 2013 tarihinden bu yana kendisine gelen ve ekinde zararlı yazılım bulunan çoğu sahte e-postayı incelememem için benimle paylaşıyor. Kendisine gönderilen e-postalara bakıldığında, 2013 yılından bu yana aktif olarak java ile zararlı yazılım geliştiren (indirici/dropper) ve sahte e-postalar gönderen bir grubun bu e-postaların arkasında olduğunu anlamak çok zor değil. Antivirüs yazılımlarını atlatmak için çeşitli gizleme (obfuscator) araçlarından da faydalanan bu grup, her salgında ikna adına aşağıdaki gibi yeni senaryolar kullanmaktan çekinmiyor.

Task sonucu bilgilendirme.

extratap.turkcell.com.tr Kişilere ekle 03.03.2015

Kime: tarik

Değerli Bayiimiz;
Aşağıda kimlik bilgileri bulunan abonemiz için açmış olduğunuz task sonuçlanmıştır.
Akışı izlemek için lütfen linke tıklayınız..
http://global-bilgi.com.tr/atachmt/tasksonucu/

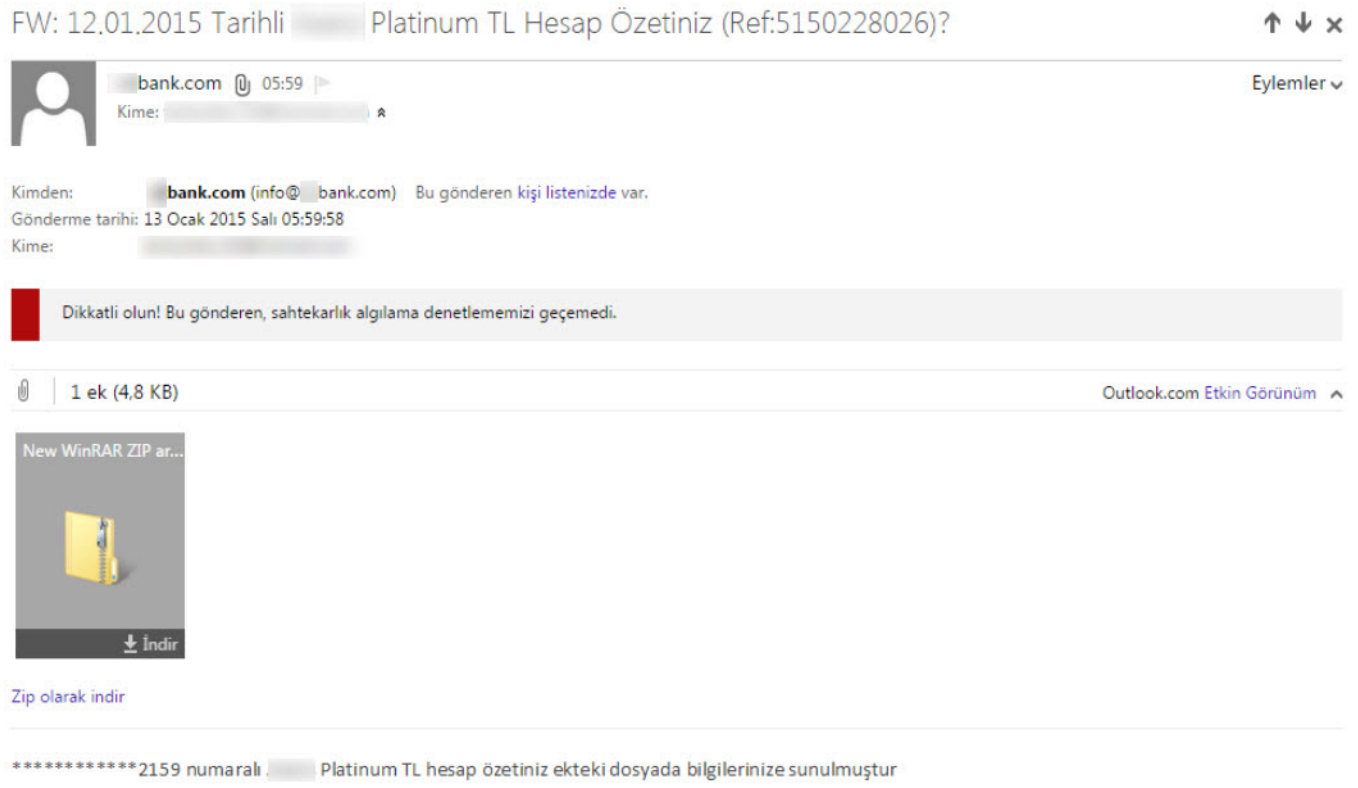
Zararlı yazılım geliştiricileri çoğunlukla indiricileri geliştirirken Java programalama dilinden faydalanıyor olmalarının en büyük sebebi, Java'nın

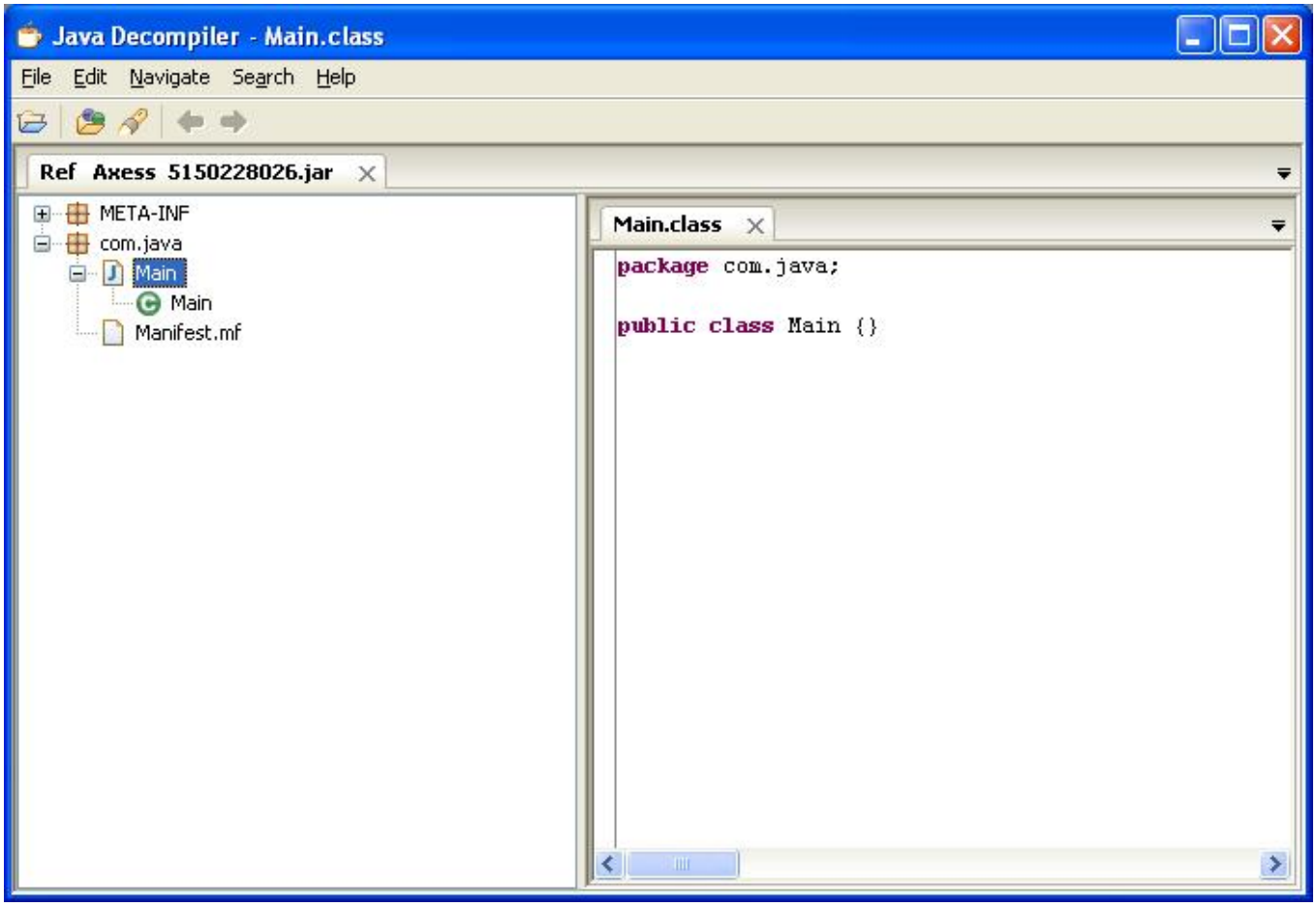
yorumlanan (interpreted) bir programlama dili olmasıdır. Bu sayede CLASS dosyasına derlenen .JAVA uzantılı bir kod, Java Virtual Machine (JVM) tarafından çalışma esnasında yorumlanarak makine diline dönüştürülmektedir. Bu durum da JAVA programlama dili ile yazılan zararlı yazılımların Antivirüs yazılımları tarafından kimi durumlarda yorumlanamamasına sebebiyet vermektedir. Ayrıca JVM tarafından çalıştırılan JAR uzantılı yürütülebilir dosyalar veya CLASS uzantılı dosyalar, Ollydbg veya Immunity Debugger gibi hata ayıklayıcılar tarafından çalıştırılamaz dolayısıyla dinamik kod analizi ile analiz edilmesi, PE dosyalara kıyasla daha zordur.

Ancak yorumlanabilir diller, diğer dillerin aksine kaynak koduna geri çevrilebilmektedir (decompile). Zararlı yazılım analistleri için ilk bakışta bu büyük bir nimet gibi görünse de, bunu bilen zararlı yazılım geliştiricileri, Allatori gibi gizleme araçlarından (obfuscator) faydalanmaktadırlar. Böyle bir durumla karşılaştığınız zaman statik bayt kod analizi ile ilerlemek iyi bir tercih gibi görünse de, dinamik bayt kod analizinin yerini zaman ve pratiklik açısından tutmayacaktır.

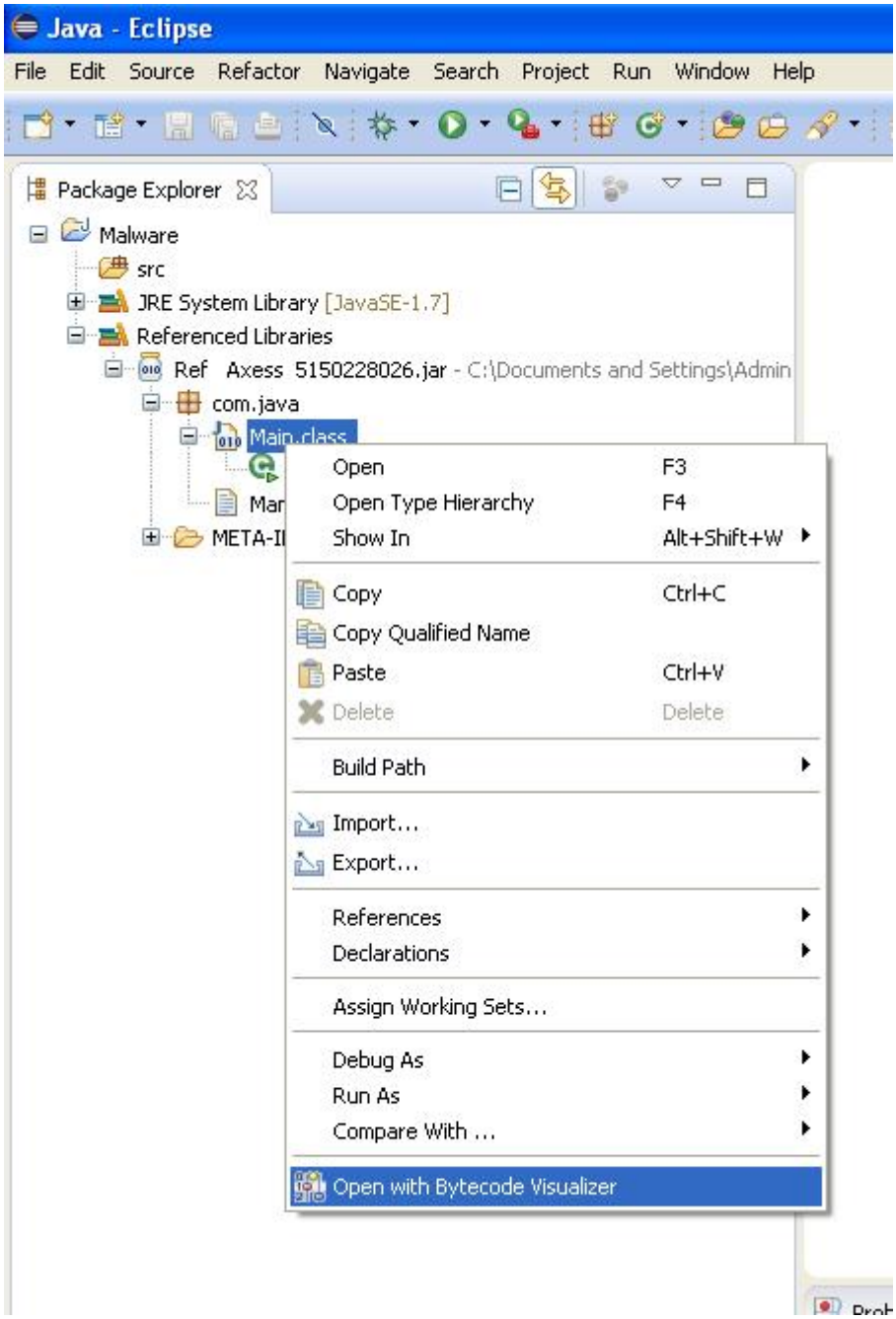
Allatori gibi araçların özelliklerine baktığınız zaman analizi güçleştiren çeşitli özellikler ile donatıldığını görebilirsiniz dolayısıyla kaynak koduna çevirme ve dinamik bayt kod analizi konusunda sıkıntı yaşayacağınız bir gerçektir!

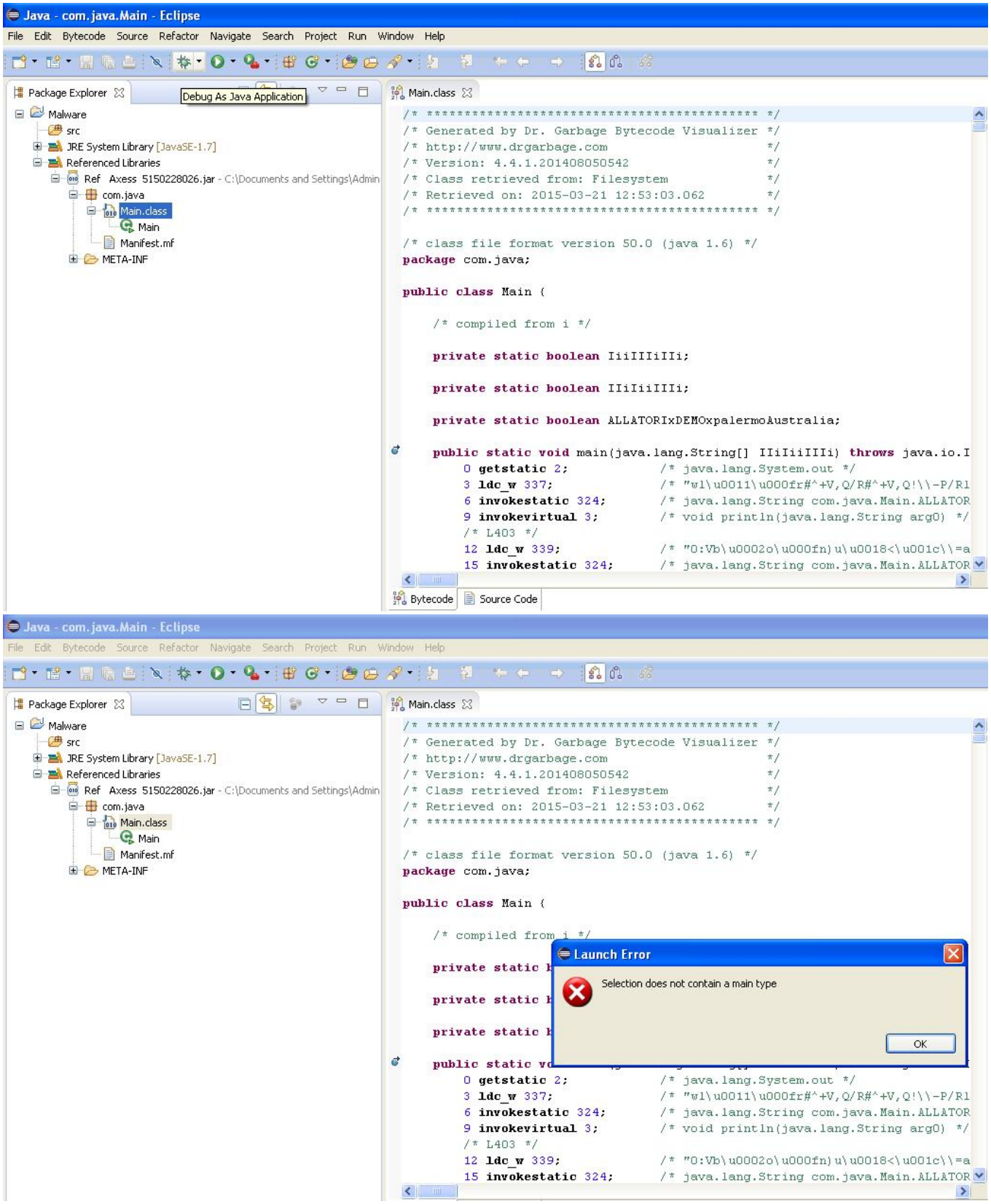
Örneğin aşağıdaki sahte e-posta ile gönderilen JAR dosyasını Java Decompiler aracı ile kaynak koduna çevirmeye çalıştığımızda kaynak kodunu görüntülemiyor olmamız bizi pek şaşırtmıyor.



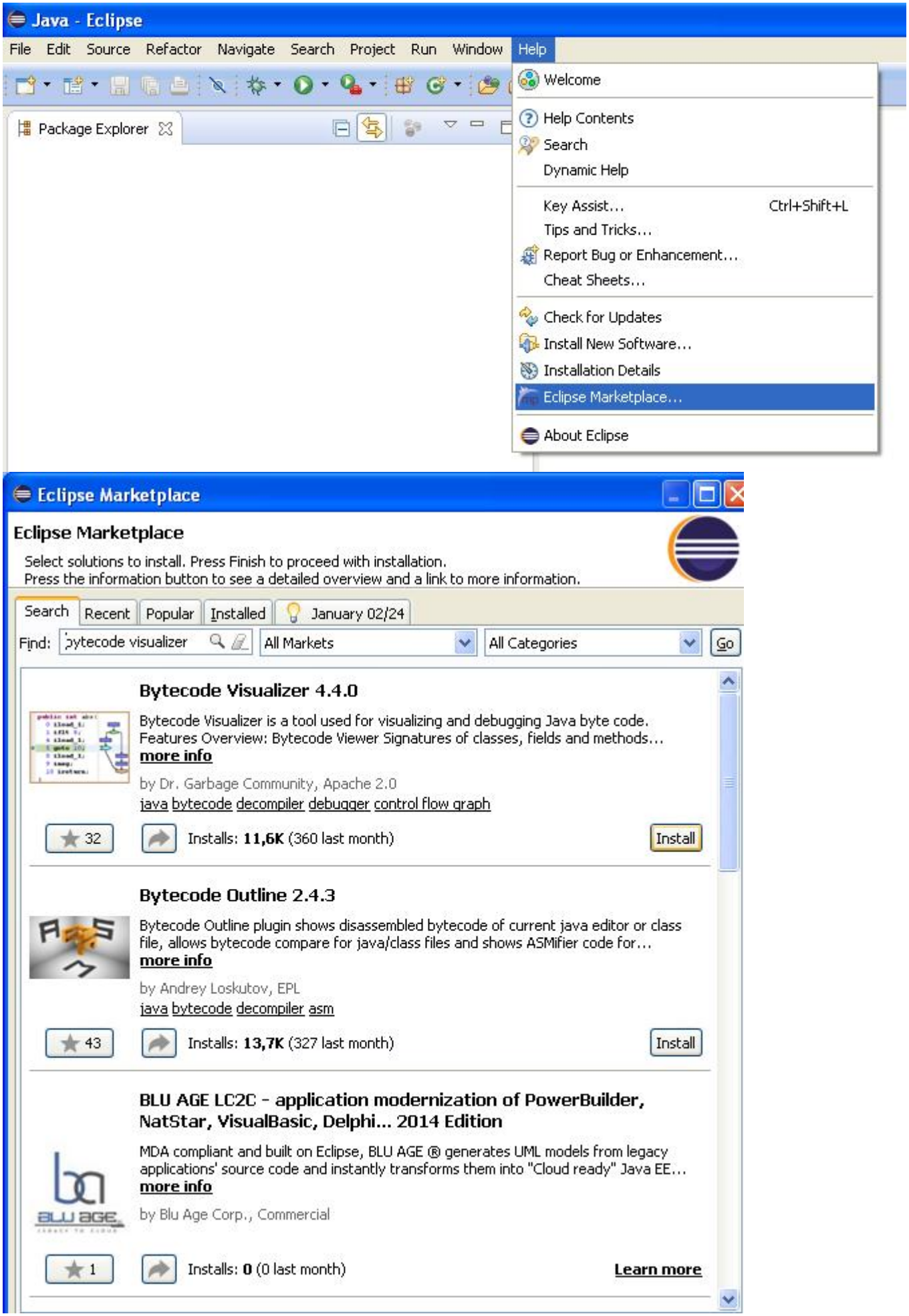


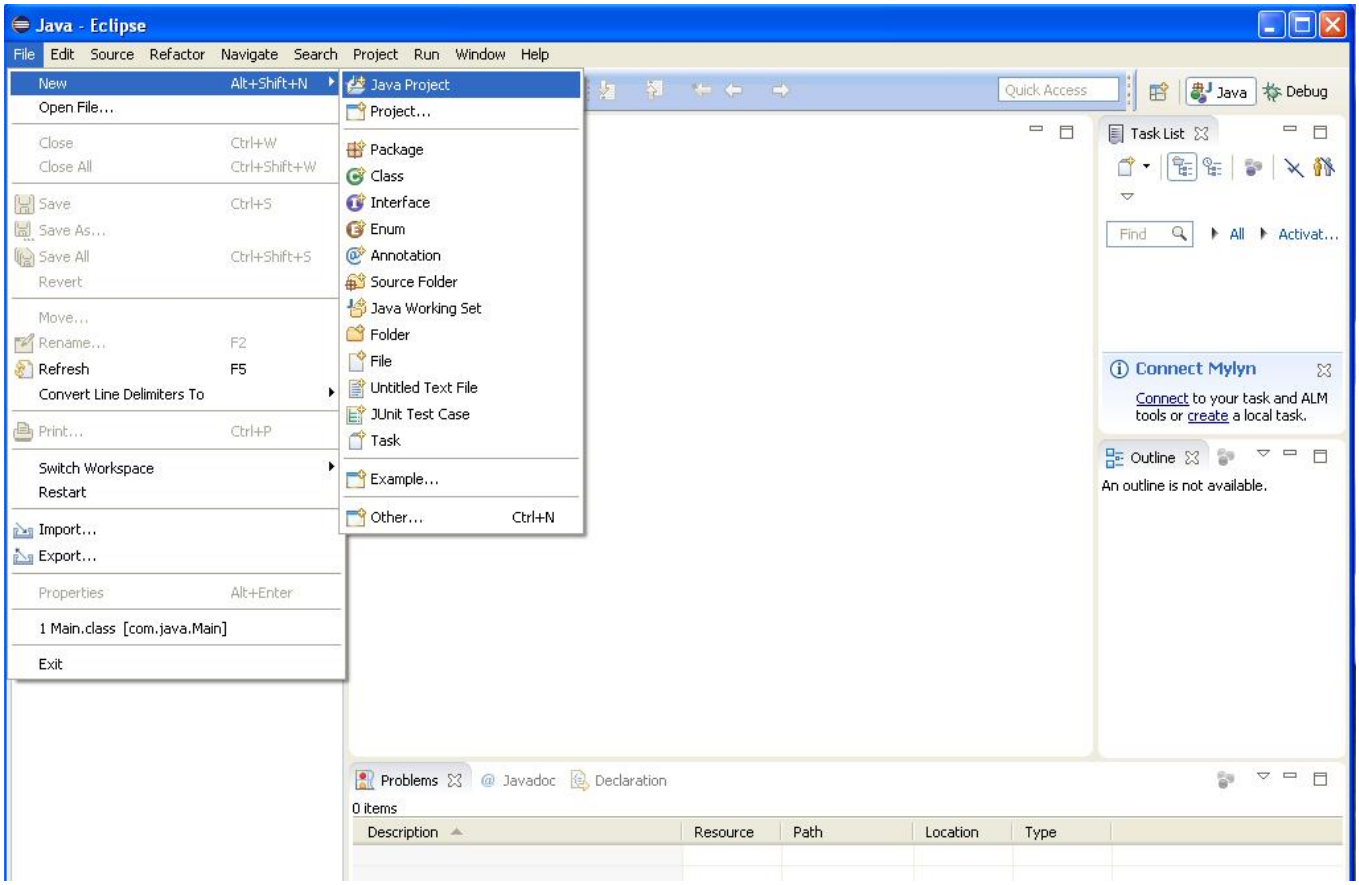
Aynı şekilde Eclipse eklentisi olarak kullanılan ve bir bayt kod hata ayıklayıcısı olan Dr. Garbage Bytecode Visualizer aracı ile de JAR dosyasını çalıştırdığımızda bir hata ile karşılaşmamız da bizi şaşırtmıyor.

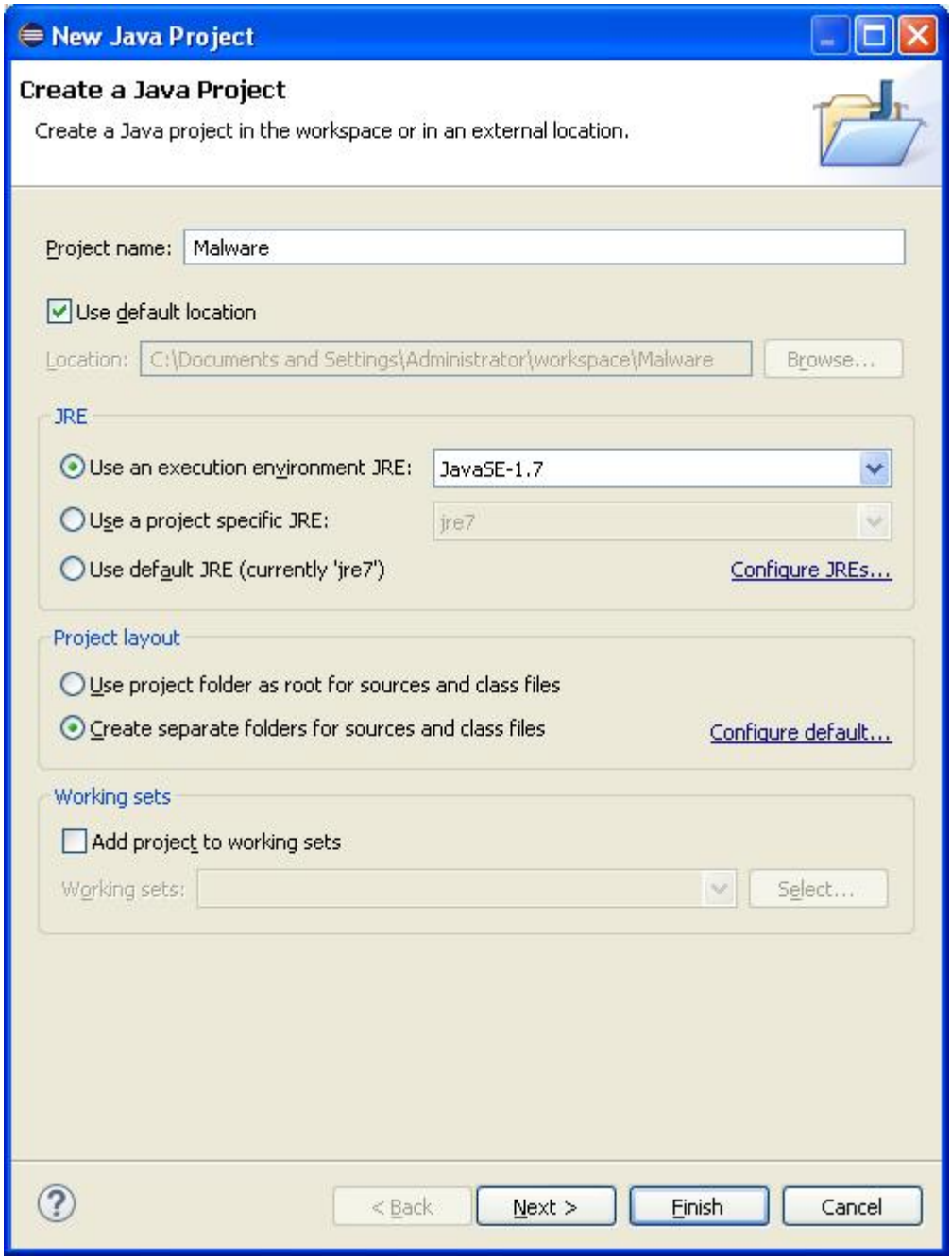

Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Malware
src
JRE System Library [JavaSE-1.7]
Referenced Libraries
Ref Axess 5150228026.jar - C:\Documents and Settings\Admin
com.java
Main.class
Open F3
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Import...
Export...
References
Declarations
Assign Working Sets...
Debug As
Run As
Compare With ...
Open with Bytecode Visualizer

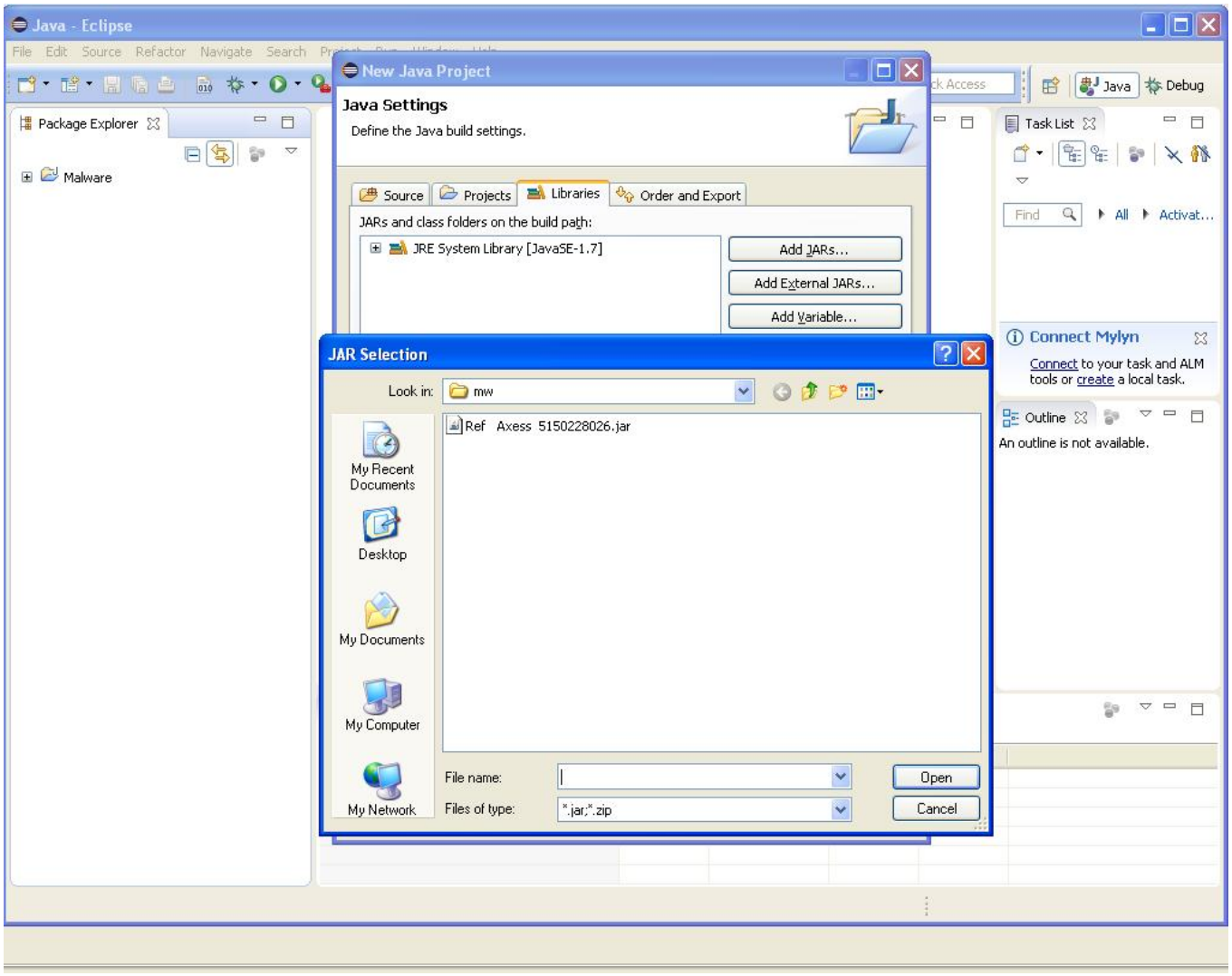


Aşağıdaki adımlardan sırasıyla geçerek hızlıca Dr. Garbage araçlarını yükleyebilir ve zararlı JAR dosyasını analiz edebilirsiniz.

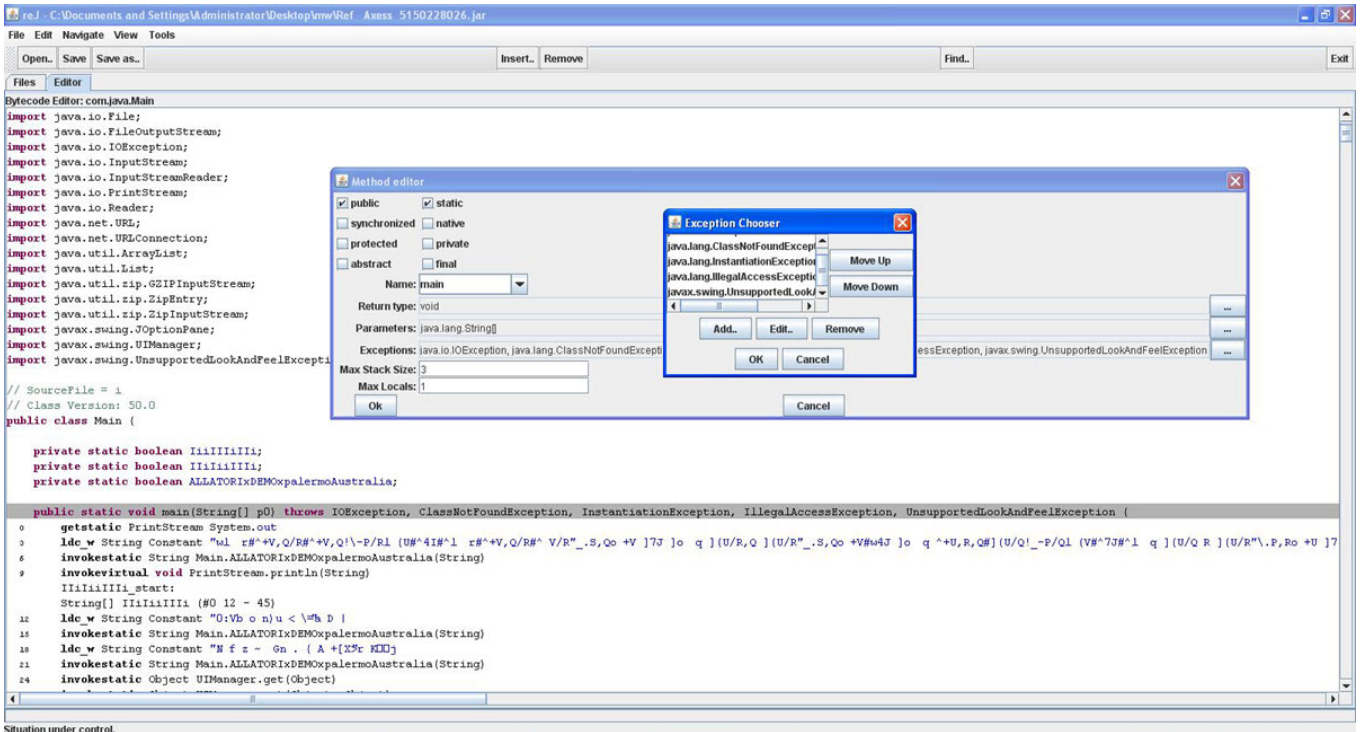

Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Welcome
Help Contents
Search
Dynamic Help
Key Assist... Ctrl+Shift+L
Tips and Tricks...
Report Bug or Enhancement...
Cheat Sheets...
Check for Updates
Install New Software...
Installation Details
Eclipse Marketplace...
About Eclipse
Eclipse Marketplace
Eclipse Marketplace
Select solutions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more information.
Search Recent Popular Installed January 02/24
Find: bytecode visualizer All Markets All Categories Go
Bytecode Visualizer 4.4.0
Bytecode Visualizer is a tool used for visualizing and debugging Java byte code.
Features Overview: Bytecode Viewer Signatures of classes, fields and methods...
more info
by Dr. Garbage Community, Apache 2.0
java bytecode decompiler debugger control flow graph
32
Installs: 11,6K (360 last month)
Install
Bytecode Outline 2.4.3
Bytecode Outline plugin shows disassembled bytecode of current java editor or class file, allows bytecode compare for java/class files and shows ASMifier code for...
more info
by Andrey Loskutov, EPL
java bytecode decompiler asm
43
Installs: 13,7K (327 last month)
Install
BLU AGE LC2C - application modernization of PowerBuilder, NatStar, VisualBasic, Delphi... 2014 Edition
MDA compliant and built on Eclipse, BLU AGE ® generates UML models from legacy applications' source code and instantly transforms them into "Cloud ready" Java EE...
more info
by Blu Age Corp., Commercial
1
Installs: 0 (0 last month)
Learn more

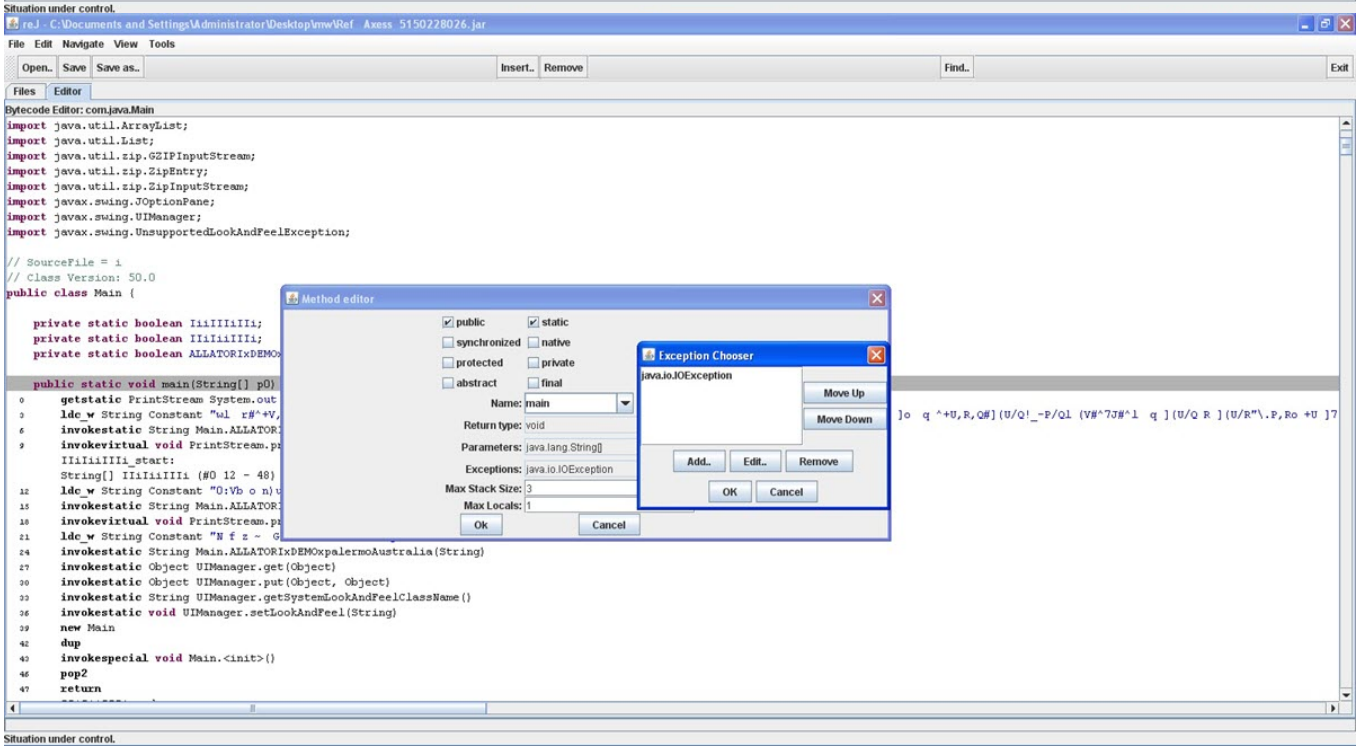

Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Restart
Import...
Export...
Properties Alt+Enter
1 Main.class [com.java.Main]
Exit
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Task
Example...
Other... Ctrl+N
Quick Access
Java
Debug
Task List
Find
All
Activat...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
An outline is not available.
Problems
Javadoc
Declaration
0 items
Description
Resource
Path
Location
Type

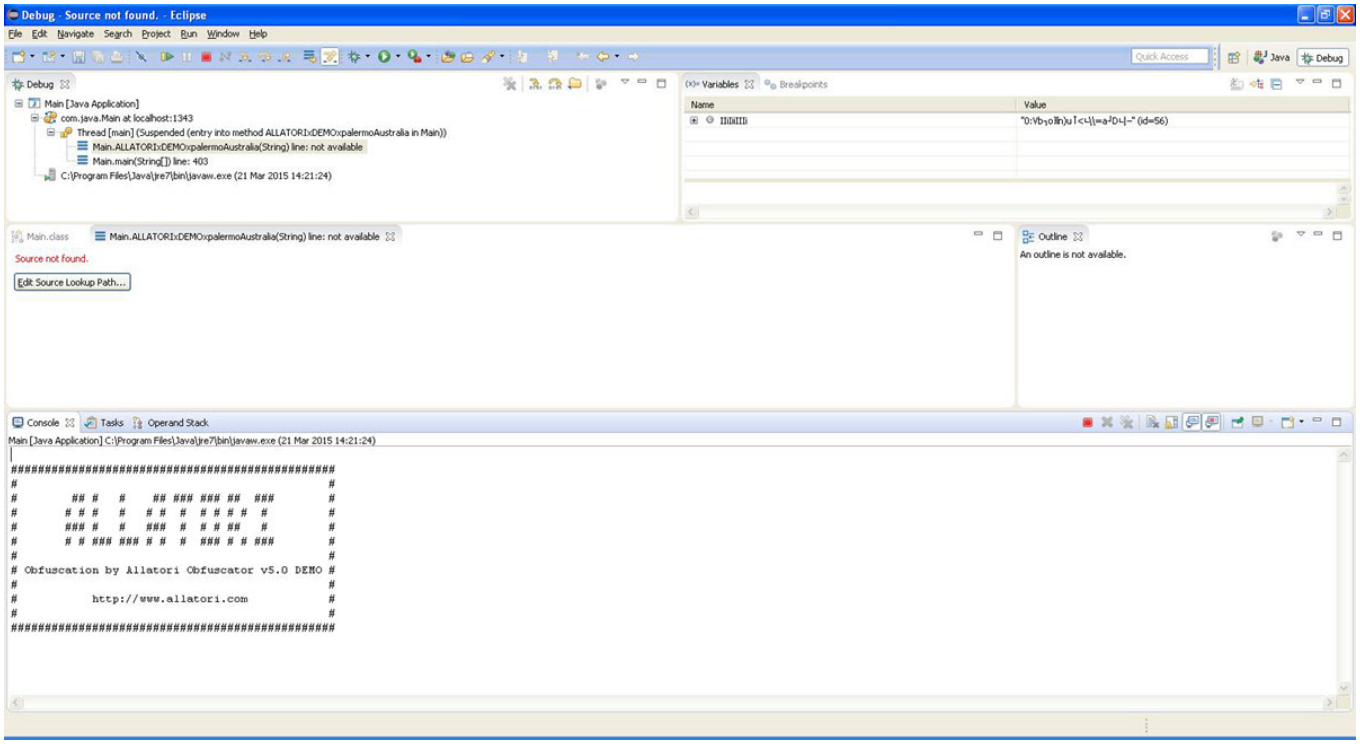

New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
Project name: Malware
Use default location
Location: C:\Documents and Settings\Administrator\workspace\Malware
Browse...
JRE
Use an execution environment JRE: JavaSE-1.7
Use a project specific JRE: jre7
Use default JRE (currently 'jre7')
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel



Bu tür analizi zorlaştırmaya yönelik yöntemlerden faydalanan Java zararlı yazılımlarına karşı reJ isimli, Java bayt kodunu manipüle etmeye imkan tanıyan araçlardan faydalanabilirsiniz. Örneğin reJ aracı ile Ref Axess 5150228026.jar dosyasını incelediğimizde Main fonksiyonunda tanımlanan çok sayıda istisnanın (exceptions) şüpheli olduğu dikkatimizi çekiyor. İstisna listesini kısaltıp kayıt ettikten sonra bu Java dosyasını başarıyla Dr. Garbage'nin bayt kod hata ayıklama aracı ile analiz edebildiğimizi görüyoruz. Bundan sonrasi ise artık ilgili yerlere kesme noktası (breakpoint) koymaya ve analiz etmeye kalıyor.


reJ - C:\Documents and Settings\Administrator\Desktop\mw\Ref Axess 5150228026.jar
File Edit Navigate View Tools
Open.. Save Save as.. Insert.. Remove Find.. Exit
Files Editor
Bytecode Editor: com.java.Main
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.io.Reader;
import java.net.URL;
import java.net.URLConnection;
import java.util.ArrayList;
import java.util.List;
import java.util.zip.GZIPInputStream;
import java.util.zip.ZipEntry;
import java.util.zip.ZipInputStream;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelExcepti
// SourceFile = i
// Class Version: 50.0
public class Main {
private static boolean IiiIIIiIIi;
private static boolean IIiIiiIIIi;
private static boolean ALLATORIxDEMOxpalermoAustralia;
public static void main(String[] p0) throws IOException, ClassNotFoundException, InstantiationException, IllegalAccessException, UnsupportedLookAndFeelException {
getstatic PrintStream System.out
ldc_w String Constant "wl r#^+V,Q/R#^+V,Q!\-P/Rl (U#^4I#^l r#^+V,Q/R#^ V/R"_.S,Qo +V]7J]o q](U/R,Q](U/R"_.S,Qo +V#w4J]o q ^+U,R,Q#](U/Q!_-P/Ql (V#^7J#^l q](U/Q R](U/R"\.P,Ro +U]7
invokestatic String Main.ALLATORIxDEMOxpalermoAustralia(String)
invokevirtual void PrintStream.println(String)
IIiIiiIIIi_start:
String[] IIiIiiIIIi (#0 12 - 45)
ldc_w String Constant "O:Vb o n)u < \=h D |
invokestatic String Main.ALLATORIxDEMOxpalermoAustralia(String)
ldc_w String Constant "N f z ~ Gn . (A +[X5r K□□j
invokestatic String Main.ALLATORIxDEMOxpalermoAustralia(String)
invokestatic Object UIManager.get(Object)
Method editor
public static
synchronized native
protected private
abstract final
Name: main
Return type: void
Parameters: java.lang.String[]
Exceptions: java.io.IOException, java.lang.ClassNotFoundExcept
Max Stack Size: 3
Max Locals: 1
Ok Cancel
Exception Chooser
Move Up
Move Down
Add.. Edit.. Remove
OK Cancel
Situation under control.


reJ - C:\Documents and Settings\Administrator\Desktop\mw\Ref Axess 5150228026.jar
File Edit Navigate View Tools
Open.. Save Save as.. Insert.. Remove Find.. Exit
Files Editor
Bytecode Editor: com.java.Main
import java.util.ArrayList;
import java.util.List;
import java.util.zip.GZIPInputStream;
import java.util.zip.ZipEntry;
import java.util.zip.ZipInputStream;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
// SourceFile = i
// Class Version: 50.0
public class Main {
private static boolean IiiIIIiIIi;
private static boolean IIiIiiIIIi;
public static void main(String[] p0)
getstatic PrintStream System.out
invokestatic String Main.ALLATORIxDEMOxpalermoAustralia(String)
invokestatic Object UIManager.get(Object)
invokestatic Object UIManager.put(Object, Object)
invokestatic String UIManager.getSystemLookAndFeelClassName()
invokestatic void UIManager.setLookAndFeel(String)
new Main
dup
invokespecial void Main.<init>()
pop2
return
Method editor
public static
synchronized native
protected private
abstract final
Name: main
Return type: void
Parameters: java.lang.String[]
Exceptions: java.io.IOException
Max Stack Size: 3
Max Locals: 1
Ok Cancel
Exception Chooser
java.io.IOException
Move Up
Move Down
Add.. Edit.. Remove
OK Cancel
Situation under control.



Java ile geliştirilen zararlı yazılımları analiz etme konusunda elinizin, kolunuzun bağlı olmadığını bilmeniz adına yazdığım bu yazı, umarım sizler için faydalı olmuştur. Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.