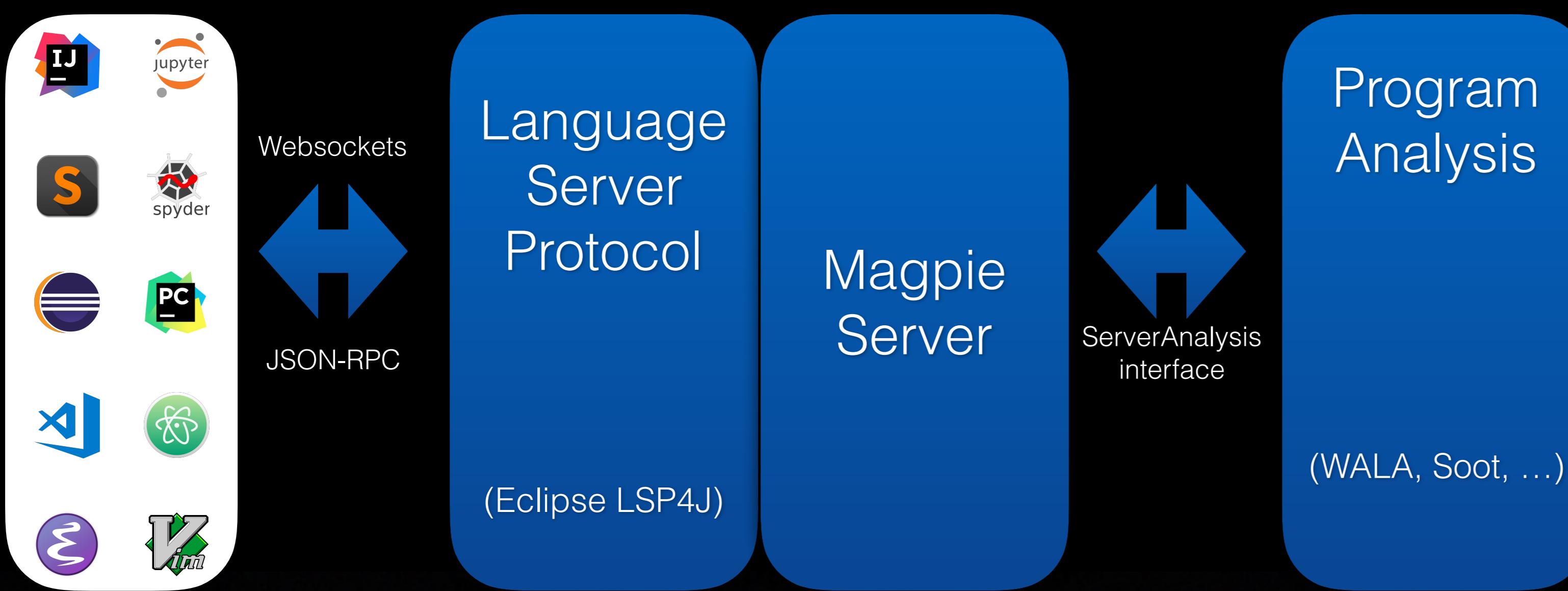


How to Use MagpieBridge: A General Approach to Integrating Static Analyses into IDEs and Editors

<https://github.com/MagpieBridge>



Additional MagpieBridge Functionalities:

- IR converter
 - adapt WALA Java IR to Soot IR
 - use WALA's precise source map
 - similar 3-address IRs
- IntelliJ LSP client
 - support popular IDE
 - Java-based
 - supports related information

Actual JavaScript Taint Analysis IDE Integration Code

```

public class CreateSDGForJavaScript {
    public static MagpieServer bridge() {
        MagpieServer bridge = new MagpieServer();
        bridge.addAnalysis("js", new ServerAnalysis() {
            @Override
            public String source() {
                return "JSTaintDemo";
            }

            @Override
            public void analyze(Collection<? extends Module> files, MagpieServer server) {
                try {
                    Collection<Module> m2 = HashSetFactory.make();
                    files.forEach(m -> m2.add(m));
                    m2.add(JSCallGraphBuilderUtil.getPrologueFile("prologue.js"));
                    Module[] modules = m2.toArray(new Module[files.size()]);
                    JavaScriptLoaderFactory loaders = new JavaScriptLoaderFactory(new CastRhinoTranslatorFactory());
                    JSCFABuilder builder = JSCFABuilderUtil.makeCGBuilder(loaders, modules, CGBuilderFactory.ONE_CFA, AstIRFactory.makeDefaultFactory());
                    CallGraph CG = builder.makeCallGraph(builder.getOptions());
                    SDGInstanceKey<SDG> SDG = new SDG<InstanceKey>(CG, builder.getPointerAnalysis(), new JavaScriptModRef<InstanceKey>(), DataDependenceOptions.NO_BASE_NO_HEAP_NO_EXCEPTIONS, ControlDependenceOptions.NONE);
                    Set<Statement> paths = Analysis.getPaths(SDG, Analysis.documentUrlSource, Analysis.documentWriteSink);
                    Set<AnalysisResult> results = HashSetFactory.make();
                    paths.forEach((path -> {
                        List<AnalysisResult> pr = new LinkedList<>();
                        for(int i = 0; i < path.size(); i++) {
                            int idx = i;
                            Statement last = path.get(idx);
                            if (Print getPosition(last) == null) {
                                continue;
                            }
                            pr.add(new AnalysisResult());
                            @Override
                            public Kind kind() {
                                return Kind.Diagnostic;
                            }

                            @Override
                            public String toString(boolean useMarkdown) {
                                if (isSink(pr)) {
                                    return "tainted sink";
                                } else if (isSource(pr)) {
                                    return "tainted source";
                                } else {
                                    return "tainted flow step";
                                }
                            }

                            private boolean isSource(List<AnalysisResult> pr) {
                                return pr.indexOf(this) == pr.size()-1;
                            }

                            private boolean isSink(List<AnalysisResult> pr) {
                                return pr.indexOf(this) == 0;
                            }

                            @Override
                            public Position position() {
                                return Print getPosition(last);
                            }

                            @Override
                            public Iterable<Pair<Position, String>> related() {
                                List<Pair<Position, String>> info = new LinkedList<>();
                                path.forEach(p -> {
                                    try {
                                        Position pos = Print getPosition(p);
                                        if (pos != null) {
                                            String text = new SourceBuffer(pos).toString();
                                            if (text != null) {
                                                info.add(Pair.of(pos, text.trim()));
                                            }
                                        }
                                    } catch (IOException e) {
                                        e.printStackTrace();
                                    }
                                });
                                Collections.reverse(info);
                                return info;
                            }

                            @Override
                            public DiagnosticSeverity severity() {
                                return isSink(pr) ? DiagnosticSeverity.Error : isSource(pr) ? DiagnosticSeverity.Warning : DiagnosticSeverity.Information;
                            }

                            @Override
                            public Pair<Position, String> repair() {
                                return null;
                            }

                            @Override
                            public String code() {
                                try {
                                    return new SourceBuffer(position()).toString();
                                } catch (IOException e) {
                                    return "unknown";
                                }
                            }
                        }
                        results.addAll(pr);
                    }));
                    server.consume(results, "JSTaintDemo");
                } catch (WalaException | IllegalArgumentException | CancelException e) {
                    MessageParams mp = new MessageParams();
                    mp.setType(MessageType.Error);
                    mp.setMessage(e.toString());
                    server.getClient().showMessage(mp);
                }
            }
        });
        return bridge;
    }

    public static void main(String[] args) throws IOException, WalaException, CancelException {
        com.ibm.wala.cast.js.ipa.callgraph.JSCallGraphUtil.setTranslatorFactory(new CastRhinoTranslatorFactory());
        MagpieServer bridge = bridge();
        bridge.launchOnStdio();
    }
}

@ServerEndpoint("/websocket-js")
public class JSTaintExampleWebsocket extends MagpieWebSocketServer {
    public JSTaintExampleWebsocket() {
        super(() -> CreateSDGForJavaScript.bridge());
        System.out.println("started server");
    }
}

```

Basic analysis code looks as usual, with little change to be suitable for the IDE. Code taken directly from WALA tutorial given in 2016*.

This analysis finds tainted flows using graph reachability on a flow- and context-sensitive system dependence graph.

1. Add analysis to MagpieServer using the `ServerAnalysis` interface

Analysis specifies applicable language
Analysis given arbitrary name for IDE

2. Adapt results using the `AnalysisResult` interface

`kind` controls how results are rendered: Diagnostics typically get underlines, and can show in any problems view
`toString` denotes the message to show, and can use markdown for some IDEs

`position` denotes where to show the message. vital that analysis frameworks be precise and accurate
`related`: analysis specifies arbitrary program items that it deems related to an issue. in this case, that is a complete tainted flow

`severity`: analysis specifies issue severity, which often controls rendered color (e.g. red for error)
`repair` denotes strings for repair by straightforward text substitution at precise code position
`code` denotes the problematic source code detected by the analysis

3. Launch MagpieServer

Websockets allows MagpieBridge to communicate with web-based systems such as Microsoft Monaco. MagpieBridge deploys seamlessly inside an Apache Tomcat server.

Monaco JS Taint Example

```

1 var document = { URL: "whatever",
2   write: function Document_prototype_write(x) { },
3   var id = document.id || (function() { return x; });
4   function Id() { this.id = id; }
5   function SubId() { }; SubId.prototype = new Id();
6
7   if (Math.random.call(null) > 0) {
8     var id1 = new Id();
9
10    [JSTaintDemo].tainted_sink
11    • temp8535751275671584956,is(9,15):document.URL
12    • temp8535751275671584956,is(10,16):id1.id.call(document,url)
13    • temp8535751275671584956,is(3,28):return x;
14    • temp8535751275671584956,is(15,1):document.write(text)
15
16

```

Monaco Python Tensors Example

```

32 def conv_net(x_dict, n_classes, dropout, reuse, is_training):
33     # Define a scope for reusing the variables
34     with tf.variable_scope('ConvNet', reuse=reuse):
35         # TF Estimator input is a dict, in case of multiple inputs
36         xxx = x_dict['images']
37
38         bad_x = tf.reshape(xxx, shape=[-1, 28, 28, 1])
39
40         # MNIST data input is a 1-D vector of 784 features (28x28 pixels)
41         # Reshape to match picture format [Height x Width x Channel]
42         # Tensor input become 4-D: [Batch Size, Height, Width, Channel]
43         z = tf.reshape(xxx, shape=[-1, 28, 28, 1])
44
45         # Convolution Layer with 32 filters and a kernel size of 5
46         conv1 = tf.layers.conv2d(z, 32, 5, activation=tf.nn.relu)
47         # Max Pooling (down-sampling) with strides of 2 and kernel size of 2
48         conv1 = tf.layers.max_pooling2d(conv1, 2, 2)
49
50         # Convolution Layer with 64 filters and a kernel size of 3
51         conv2 = tf.layers.conv2d(conv1, 64, 3, activation=tf.nn.relu)
52         # Max Pooling (down-sampling) with strides of 2 and kernel size of 2
53         conv2 = tf.layers.max_pooling2d(conv2, 2, 2)
54
55         # Convolution Layer with 128 filters and a kernel size of 3
56         conv3 = tf.layers.conv2d(conv2, 128, 3, activation=tf.nn.relu)
57         # Max Pooling (down-sampling) with strides of 2 and kernel size of 2
58         conv3 = tf.layers.max_pooling2d(conv3, 2, 2)
59

```

Fix: Replace it with `tf.reshape(xxx, [-1, 28, 28, 1])`
Report it as false alarm (Bad type to convolve pixel[n]: [28 * 28], needs 4 dimensions (possible fix: `tf.reshape(xxx, [-1, 28, 28, 1])`)).

IntelliJ CogniCrypt Example

```

1 import ...
2
3 public class RSA {
4     private PublicKey publicKey;
5     private PrivateKey privateKey;
6
7     public RSA() throws NoSuchAlgorithmException {
8         KeyPairGenerator generator = KeyPairGenerator.getInstance("RSA");
9         generator.initialize(1024);
10    th: Report at false alarm! First parameter (with value 512) should be any of (2048, 4096).
11    by itself update message();
12    instance.update(bytes);
13    instance.sign();
14    return bytes;
15
16
17    public byte[] sign(String message) throws InvalidKeyException, NoSuchAlgorithmException, SignatureException {
18        Signature signature = Signature.getInstance("SHA256WithRSA");
19        byte[] bytes = message.getBytes();
20        instance.update(bytes);
21        instance.update(message);
22        instance.update(bytes);
23        instance.sign();
24        return bytes;
25
26    }
27
28    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
29        RSA rsa = RSA();
30        rsa.sign("Hello World!");
31    }
32
33    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
34        RSA rsa = RSA();
35        rsa.sign("Hello World!");
36    }
37
38    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
39        RSA rsa = RSA();
40        rsa.sign("Hello World!");
41    }
42
43    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
44        RSA rsa = RSA();
45        rsa.sign("Hello World!");
46    }
47
48    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
49        RSA rsa = RSA();
50        rsa.sign("Hello World!");
51    }
52
53    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
54        RSA rsa = RSA();
55        rsa.sign("Hello World!");
56    }
57
58    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
59        RSA rsa = RSA();
60        rsa.sign("Hello World!");
61    }
62
63    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
64        RSA rsa = RSA();
65        rsa.sign("Hello World!");
66    }
67
68    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
69        RSA rsa = RSA();
70        rsa.sign("Hello World!");
71    }
72
73    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
74        RSA rsa = RSA();
75        rsa.sign("Hello World!");
76    }
77
78    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
79        RSA rsa = RSA();
80        rsa.sign("Hello World!");
81    }
82
83    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
84        RSA rsa = RSA();
85        rsa.sign("Hello World!");
86    }
87
88    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
89        RSA rsa = RSA();
90        rsa.sign("Hello World!");
91    }
92
93    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
94        RSA rsa = RSA();
95        rsa.sign("Hello World!");
96    }
97
98    public static void main(String[] args) throws IOException, NoSuchAlgorithmException, InvalidKeyException {
99        RSA rsa = RSA();
100       rsa.sign("Hello World!");
101
102      th: Report at false alarm! First parameter (with value 512) should be any of (2048, 4096).
103      by itself update message();
104      instance.update(bytes);
105      instance.sign();
106      return bytes;
107
108  RSA > RSA

```

Visual Studio Code FlowDroid Example

```

1 import ...
2
3 public class Database {
4     private Connection connection;
5
6     public Database() {
7         connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/test");
8     }
9
10    public void insert(String query) {
11        try {
12            Statement statement = connection.createStatement();
13            statement.executeUpdate(query);
14        } catch (SQLException e) {
15            e.printStackTrace();
16        }
17    }
18
19    public void update(String query) {
20        try {
21            Statement statement = connection.createStatement();
22            statement.executeUpdate(query);
23        } catch (SQLException e) {
24            e.printStackTrace();
25        }
26    }
27
28    public void delete(String query) {
29        try {
30            Statement statement = connection.createStatement();
31            statement.executeUpdate(query);
32        } catch (SQLException e) {
33            e.printStackTrace();
34        }
35    }
36
37    public ResultSet select(String query) {
38        try {
39            Statement statement = connection.createStatement();
40            ResultSet resultSet = statement.executeQuery(query);
41            return resultSet;
42        } catch (SQLException e) {
43            e.printStackTrace();
44        }
45    }
46
47    public void close() {
48        try {
49            connection.close();
50        } catch (SQLException e) {
51            e.printStackTrace();
52        }
53    }
54
55    public void setAutoCommit(boolean autoCommit) {
56        connection.setAutoCommit(autoCommit);
57    }
58
59    public void commit() {
60        connection.commit();
61    }
62
63    public void rollback() {
64        connection.rollback();
65    }
66
67    public void setTransactionIsolation(int level) {
68        connection.setTransactionIsolation(level);
69    }
70
71    public void setCharacterEncoding(String encoding) {
72        connection.setCharacterEncoding(encoding);
73    }
74
75    public void setSchema(String schema) {
76        connection.setSchema(schema);
77    }
78
79    public void setHoldability(int holdability) {
80        connection.setHoldability(holdability);
81    }
82
83    public void setCatalog(String catalog) {
84        connection.setCatalog(catalog);
85    }
86
87    public void setReadOnly(boolean readOnly) {
88        connection.setReadOnly(readOnly);
89    }
90
91    public void setXACT_ABORT(boolean xactAborting) {
92        connection.setXACT_ABORT(xactAborting);
93    }
94
95    public void setXACT重生(boolean xactResuming) {
96        connection.setXACT重生(xactResuming);
97    }
98
99    public void setXACT重生(boolean xactResuming) {
100       connection.setXACT重生(xactResuming);
101
102      th: Report at false alarm! First parameter (with value 512) should be any of (2048, 4096).
103      by itself update message();
104      instance.update(bytes);
105      instance.sign();
106      return bytes;
107
108  Database > Database

```

*cited tutorial: <https://conf.researchr.org/track/pldi-2016/Tutorials>