RemoteFlowSource类

使用类

在codeql中常见一个类，那就是remote flow source类，这个类常见的使用方法是：

```
override predicate isSource(DataFlow::Node source) {
    source instanceof RemoteFlowSource
   }
```

类原型

一般多见于source的描述和定义。他的原型是在flowsources.qll中

```
/**
 * Provides classes representing various flow sources for taint
 tracking.
 */

import java
import semmle.code.java.dataflow.DataFlow
import semmle.code.java.dataflow.TaintTracking
import semmle.code.java.dataflow.DefUse
import semmle.code.java.frameworks.Jdbc
import semmle.code.java.frameworks.Networking
import semmle.code.java.frameworks.Properties
import semmle.code.java.frameworks.Rmi
import semmle.code.java.frameworks.Servlets
import semmle.code.java.frameworks.ApacheHttp
import semmle.code.java.frameworks.android.XmlParsing
import semmle.code.java.frameworks.android.WebView
import semmle.code.java.frameworks.JaxWS
```

```
import semmle.code.java.frameworks.android.Intent
import semmle.code.java.frameworks.spring.SpringWeb
import semmle.code.java.frameworks.spring.SpringController
import semmle.code.java.frameworks.spring.SpringWebClient
import semmle.code.java.frameworks.Guice
import semmle.code.java.frameworks.struts.StrutsActions
import semmle.code.java.frameworks.Thrift

/** A data flow source of remote user input. */
abstract class RemoteFlowSource extends DataFlow::Node {
  /** Gets a string that describes the type of this remote flow source. */
  abstract string getSourceType();
}

private class RemoteTaintedMethodAccessSource extends RemoteFlowSource {
  RemoteTaintedMethodAccessSource() {
    this.asExpr().(MethodAccess).getMethod() instanceof RemoteTaintedMethod
  }

  override string getSourceType() { result = "network data source" }
}

private class RmiMethodParameterSource extends RemoteFlowSource {
  RmiMethodParameterSource() {
    exists(RemoteCallableMethod method |
      method.getAParameter() = this.asParameter() and
      (
        getType() instanceof PrimitiveType or
        getType() instanceof TypeString
      )
    )
  }

  override string getSourceType() { result = "RMI method parameter" }
```

```
52 }
53
54 private class JaxWsMethodParameterSource extends RemoteFlowSourc
   e {
55   JaxWsMethodParameterSource() {
56     exists(JaxWsEndpoint endpoint |
57       endpoint.getARemoteMethod().getAParameter() = this.asParam
   eter()
58     )
59   }
60
61   override string getSourceType() { result = "Jax WS method para
   meter" }
62 }
63
64 private class JaxRsMethodParameterSource extends RemoteFlowSourc
   e {
65   JaxRsMethodParameterSource() {
66     exists(JaxRsResourceClass service |
67       service.getAnInjectableCallable().getAParameter() = this.a
   sParameter() or
68       service.getAnInjectableField().getAnAccess() = this.asExpr
   ()
69     )
70   }
71
72   override string getSourceType() { result = "Jax Rs method para
   meter" }
73 }
74
75 private predicate variableStep(Expr tracked, VarAccess sink) {
76   exists(VariableAssign def |
77     def.getSource() = tracked and
78     defUsePair(def, sink)
79   )
80 }
81
82 private class ReverseDnsSource extends RemoteFlowSource {
83   ReverseDnsSource() {
84     // Try not to trigger on `localhost`.
```

```
85      exists(MethodAccess m | m = this.asExpr() |
86        m.getMethod() instanceof ReverseDNSMethod and
87        not exists(MethodAccess l |
88          (variableStep(l, m.getQualifier()) or l = m.getQualifier
    ()) and
89          l.getMethod().getName() = "getLocalHost"
90        )
91      )
92    }
93
94    override string getSourceType() { result = "reverse DNS looku
    p" }
95  }
96
97  private class MessageBodyReaderParameterSource extends RemoteFlo
    wSource {
98    MessageBodyReaderParameterSource() {
99      exists(MessageBodyReaderRead m |
100       m.getParameter(4) = this.asParameter() or
101       m.getParameter(5) = this.asParameter()
102     )
103   }
104
105   override string getSourceType() { result = "MessageBodyReader
     parameter" }
106 }
107
108 private class SpringMultipartFileSource extends RemoteFlowSource
    {
109   SpringMultipartFileSource() {
110     exists(MethodAccess ma, Method m |
111       ma = this.asExpr() and
112       m = ma.getMethod() and
113       m.getDeclaringType().hasQualifiedName("org.springframewor
    k.web.multipart", "MultipartFile") and
114       m.getName().matches("get%")
115     )
116   }
117
118   override string getSourceType() { result = "Spring MultipartFi
```

```
    le getter" }
119 }
120
121 private class SpringServletInputParameterSource extends RemoteFl
    owSource {
122   SpringServletInputParameterSource() {
123     this.asParameter() = any(SpringRequestMappingParameter srmp
    | srmp.isTaintedInput())
124   }
125
126   override string getSourceType() { result = "Spring servlet inp
    ut parameter" }
127 }
128
129 private class GuiceRequestParameterSource extends RemoteFlowSour
    ce {
130   GuiceRequestParameterSource() {
131     exists(GuiceRequestParametersAnnotation a |
132       a = this.asParameter().getAnAnnotation() or
133       a = this.asExpr().(FieldRead).getField().getAnAnnotation()
134     )
135   }
136
137   override string getSourceType() { result = "Guice request para
    meter" }
138 }
139
140 private class Struts2ActionSupportClassFieldReadSource extends R
    emoteFlowSource {
141   Struts2ActionSupportClassFieldReadSource() {
142     exists(Struts2ActionSupportClass c |
143       c.getASetterMethod().getField() = this.asExpr().(FieldRead
    ).getField()
144     )
145   }
146
147   override string getSourceType() { result = "Struts2 ActionSupp
    ort field" }
148 }
149
```

```
150 private class ThriftIfaceParameterSource extends RemoteFlowSourc
    e {
151   ThriftIfaceParameterSource() {
152     exists(ThriftIface i | i.getAnImplementingMethod().getAParam
    eter() = this.asParameter())
153   }
154
155   override string getSourceType() { result = "Thrift Iface param
    eter" }
156 }
157
158 /** Class for `tainted` user input. */
159 abstract class UserInput extends DataFlow::Node { }
160
161 /**
162  * DEPRECATED: Use `RemoteFlowSource` instead.
163  *
164  * Input that may be controlled by a remote user.
165  */
166 deprecated class RemoteUserInput extends UserInput {
167   RemoteUserInput() { this instanceof RemoteFlowSource }
168 }
169
170 /** A node with input that may be controlled by a local user. */
171 abstract class LocalUserInput extends UserInput { }
172
173 /**
174  * A node with input from the local environment, such as files,
     standard in,
175  * environment variables, and main method parameters.
176  */
177 class EnvInput extends LocalUserInput {
178   EnvInput() {
179     // Parameters to a main method.
180     exists(MainMethod main | this.asParameter() = main.getParame
    ter(0))
181     or
182     // Args4j arguments.
183     exists(Field f | this.asExpr() = f.getAnAccess() |
184       f.getAnAnnotation().getType().getQualifiedName() = "org.ko
```

```
hsuke.args4j.Argument"
185     )
186     or
187     // Results from various specific methods.
188     this.asExpr().(MethodAccess).getMethod() instanceof EnvTaint
edMethod
189     or
190     // Access to `System.in`.
191     exists(Field f | this.asExpr() = f.getAnAccess() | f instanc
eof SystemIn)
192     or
193     // Access to files.
194     this
195         .asExpr()
196         .(ConstructorCall)
197         .getConstructedType()
198         .hasQualifiedName("java.io", "FileInputStream")
199   }
200 }
201
202 /** A node with input from a database. */
203 class DatabaseInput extends LocalUserInput {
204   DatabaseInput() { this.asExpr().(MethodAccess).getMethod() ins
tanceof ResultSetGetStringMethod }
205 }
206
207 private class RemoteTaintedMethod extends Method {
208   RemoteTaintedMethod() {
209     this instanceof ServletRequestGetParameterMethod or
210     this instanceof ServletRequestGetParameterMapMethod or
211     this instanceof ServletRequestGetParameterNamesMethod or
212     this instanceof HttpServletRequestGetQueryStringMethod or
213     this instanceof HttpServletRequestGetHeaderMethod or
214     this instanceof HttpServletRequestGetPathMethod or
215     this instanceof HttpServletRequestGetHeadersMethod or
216     this instanceof HttpServletRequestGetHeaderNamesMethod or
217     this instanceof HttpServletRequestGetRequestURIMethod or
218     this instanceof HttpServletRequestGetRequestURLMethod or
219     this instanceof HttpServletRequestGetRemoteUserMethod or
220     this instanceof SpringWebRequestGetMethod or
```

```
221     this instanceof SpringRestTemplateResponseEntityMethod or
222     this instanceof ServletRequestGetBodyMethod or
223     this instanceof CookieGetValueMethod or
224     this instanceof CookieGetNameMethod or
225     this instanceof CookieGetCommentMethod or
226     this instanceof URLConnectionGetInputStreamMethod or
227     this instanceof SocketGetInputStreamMethod or
228     this instanceof ApacheHttpGetParams or
229     this instanceof ApacheHttpEntityGetContent or
230     // In the setting of Android we assume that XML has been tra
    nsmitted over
231     // the network, so may be tainted.
232     this instanceof XmlPullGetMethod or
233     this instanceof XmlAttrSetGetMethod or
234     // The current URL in a browser may be untrusted or uncontro
    lled.
235     this instanceof WebViewGetUrlMethod
236   }
237 }
238
239 private class SpringWebRequestGetMethod extends Method {
240   SpringWebRequestGetMethod() {
241     exists(SpringWebRequest swr | this = swr.getAMethod() |
242       this.hasName("getDescription") or
243       this.hasName("getHeader") or
244       this.hasName("getHeaderNames") or
245       this.hasName("getHeaderValues") or
246       this.hasName("getParameter") or
247       this.hasName("getParameterMap") or
248       this.hasName("getParameterNames") or
249       this.hasName("getParameterValues")
250       // TODO consider getRemoteUser
251     )
252   }
253 }
254
255 private class EnvTaintedMethod extends Method {
256   EnvTaintedMethod() {
257     this instanceof MethodSystemGetenv or
258     this instanceof PropertiesGetPropertyMethod or
```

```
259     this instanceof MethodSystemGetProperty
260   }
261 }
262
263 /** The type `java.net.InetAddress`. */
264 class TypeInetAddr extends RefType {
265   TypeInetAddr() { this.getQualifiedName() = "java.net.InetAddre
    ss" }
266 }
267
268 /** A reverse DNS method. */
269 class ReverseDNSMethod extends Method {
270   ReverseDNSMethod() {
271     this.getDeclaringType() instanceof TypeInetAddr and
272     (
273       this.getName() = "getHostName" or
274       this.getName() = "getCanonicalHostName"
275     )
276   }
277 }
278
279 /** Android `Intent` that may have come from a hostile applicati
    on. */
280 class AndroidIntentInput extends DataFlow::Node {
281   AndroidIntentInput() {
282     exists(MethodAccess ma, AndroidGetIntentMethod m |
283       ma.getMethod().overrides*(m) and
284       this.asExpr() = ma
285     )
286     or
287     exists(Method m, AndroidReceiveIntentMethod rI |
288       m.overrides*(rI) and
289       this.asParameter() = m.getParameter(1)
290     )
291   }
292 }
```

你可以通过继承抽象方法来扩充你所希望描述的source，通常情况下我不会去修改这个类，因为他已经比较完善了，为了更好的管理你的规则，我建议你继承部分需要的类然后再使用，而不建议修改底层的类。

这个类已经基本上涵盖了主流的java框架的获取参数方法。
其中的result会返回一个对应的变量，相当于this关键字。