David Rook

Agnitio
Security code review swiss army knife

Hack in Paris, Paris

if (slide == introduction) System.out.println("I’m David Rook");

- Security Analyst, Realex Payments, Ireland
  CISSP, CISA, GCIH and many other acronyms
- Security Ninja (www.securityninja.co.uk)
- Speaker at international security conferences
- Nominated for multiple blog awards
- A mentor in the InfoSecMentors project
- Developed and released Agnitio

The real time payment exchange

Agenda

- What is static analysis?
- Security code reviews: the good, the bad and the ugly
- The principles of secure development
- Agnitio: It’s static analysis, but not as we know it
- A sneak preview of Agnitio v2.0

SECURITY

Static analysis

- What do I mean by static analysis?
  - A review of source code without executing the application
  - Can be either manual or automated through one or more tools
  - Human and/or tools analysing application source code

realex
The real time payment exchange

SECURITY

Static analysis

- Wetware or software?
  - Humans are needed with or without static analysis tools
  - The best thing about humans is that they aren’t software

realex
The real time payment exchange

SECURITY

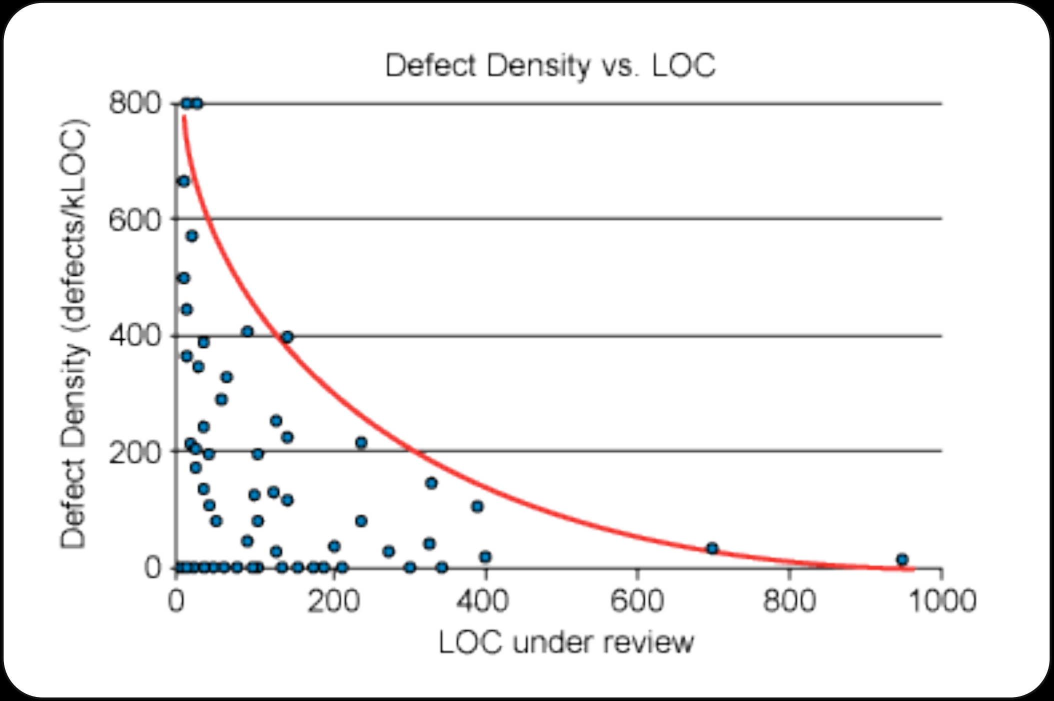
Static analysis

- Wetware or software?
  - Humans are needed with or without static analysis tools
  - The best thing about humans is that they aren't software
  - The worst thing about humans is that they are humans

realex
The real time payment exchange

Static analysis

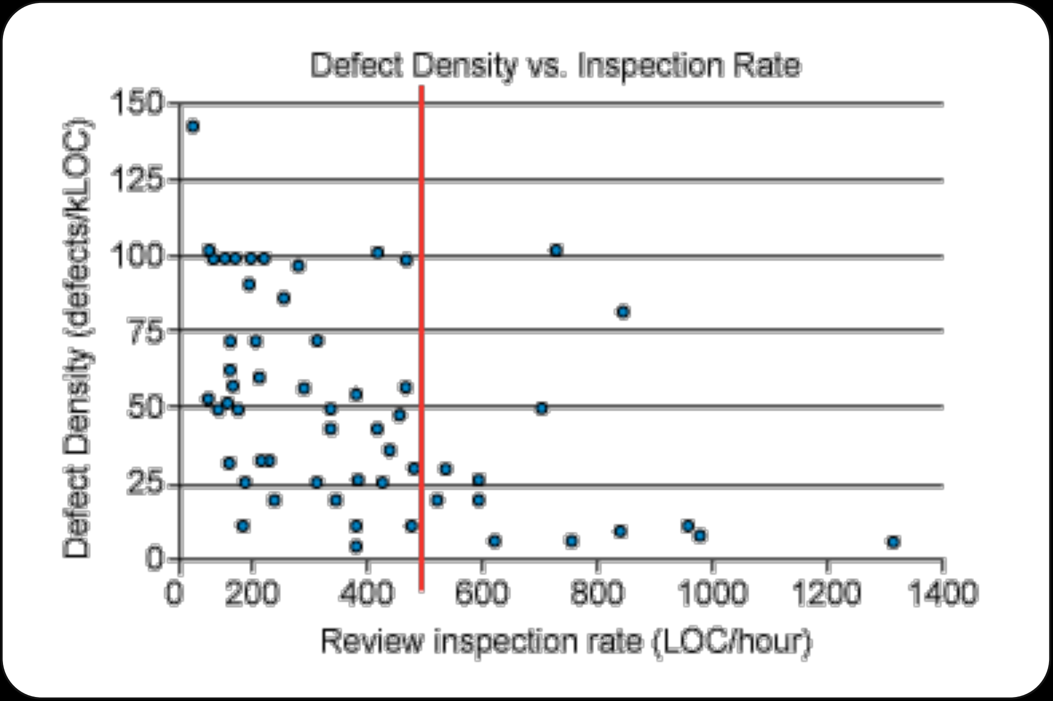
- Wetware or software?



http://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/index.html?sf1100063=1

Static analysis

- Wetware or software?



http://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/index.html?sf1100063=1

realex
The real time payment exchange

Static analysis

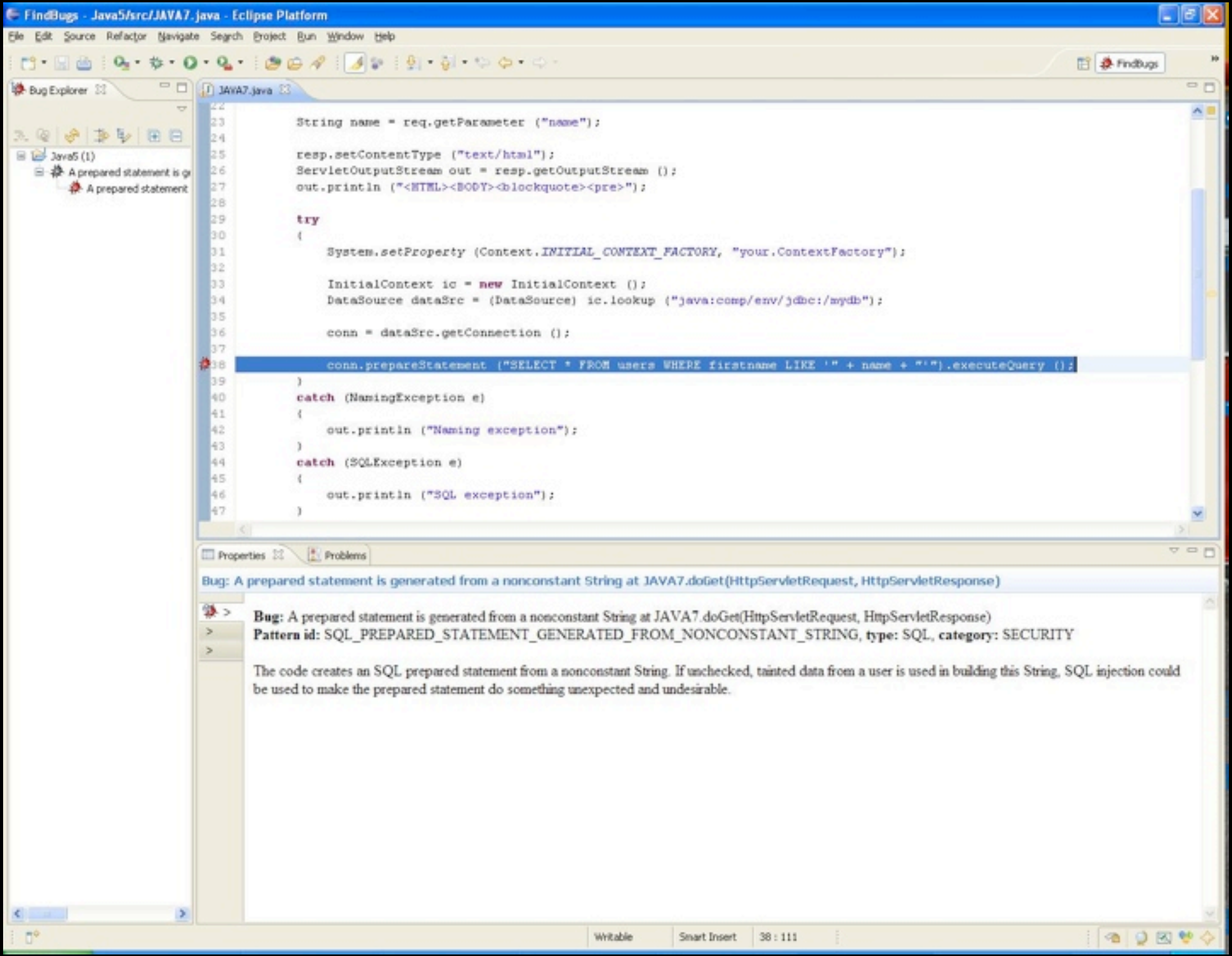
- Wetware or software?

    - Tools can cover more code in less time than a human
    - The best thing about software is that it isn't human

realex
The real time payment exchange

Static analysis

- Wetware or software?
  - Tools can cover more code in less time than a human
  - The best thing about software is that it isn’t human
  - The worst thing about software is that it’s software

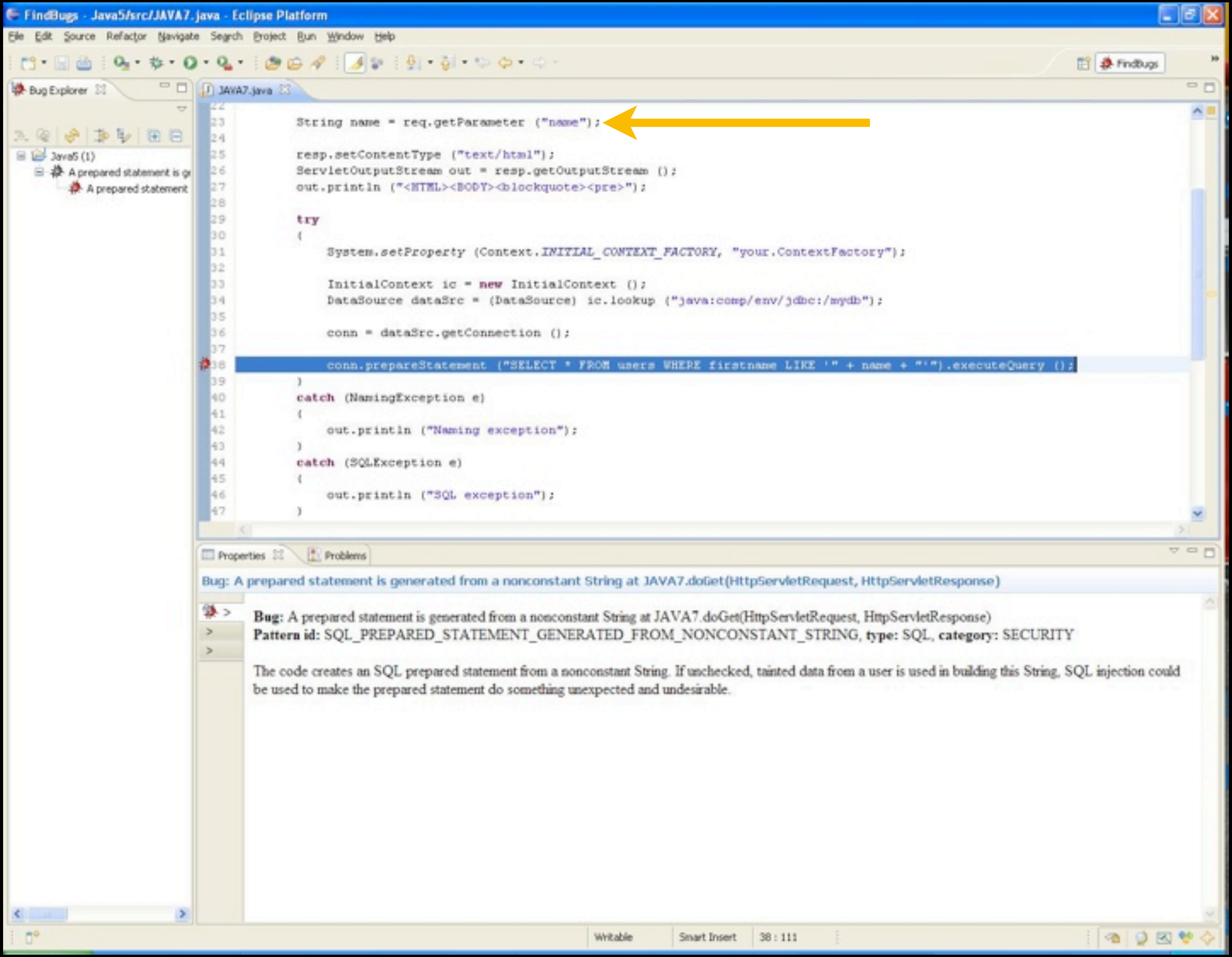

FindBugs - Java5/src/JAVA7.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
FindBugs
Bug Explorer
Java5 (1)
A prepared statement is g
A prepared statement
JAVA7.java
Properties
Problems
Writable
Smart Insert
38 : 111

```
        String name = req.getParameter ("name");

        resp.setContentType ("text/html");
        ServletOutputStream out = resp.getOutputStream ();
        out.println ("<HTML><BODY><blockquote><pre>");

        try
        {
            System.setProperty (Context.INITIAL_CONTEXT_FACTORY, "your.ContextFactory");

            InitialContext ic = new InitialContext ();
            DataSource dataSrc = (DataSource) ic.lookup ("java:comp/env/jdbc:/mydb");

            conn = dataSrc.getConnection ();

            conn.prepareStatement ("SELECT * FROM users WHERE firstname LIKE '" + name + "'").executeQuery ();
        }
        catch (NamingException e)
        {
            out.println ("Naming exception");
        }
        catch (SQLException e)
        {
            out.println ("SQL exception");
        }
```

Bug: A prepared statement is generated from a nonconstant String at JAVA7.doGet(HttpServletRequest, HttpServletResponse)

Bug: A prepared statement is generated from a nonconstant String at JAVA7.doGet(HttpServletRequest, HttpServletResponse)
Pattern id: SQL_PREPARED_STATEMENT_GENERATED_FROM_NONCONSTANT_STRING, **type:** SQL, **category:** SECURITY

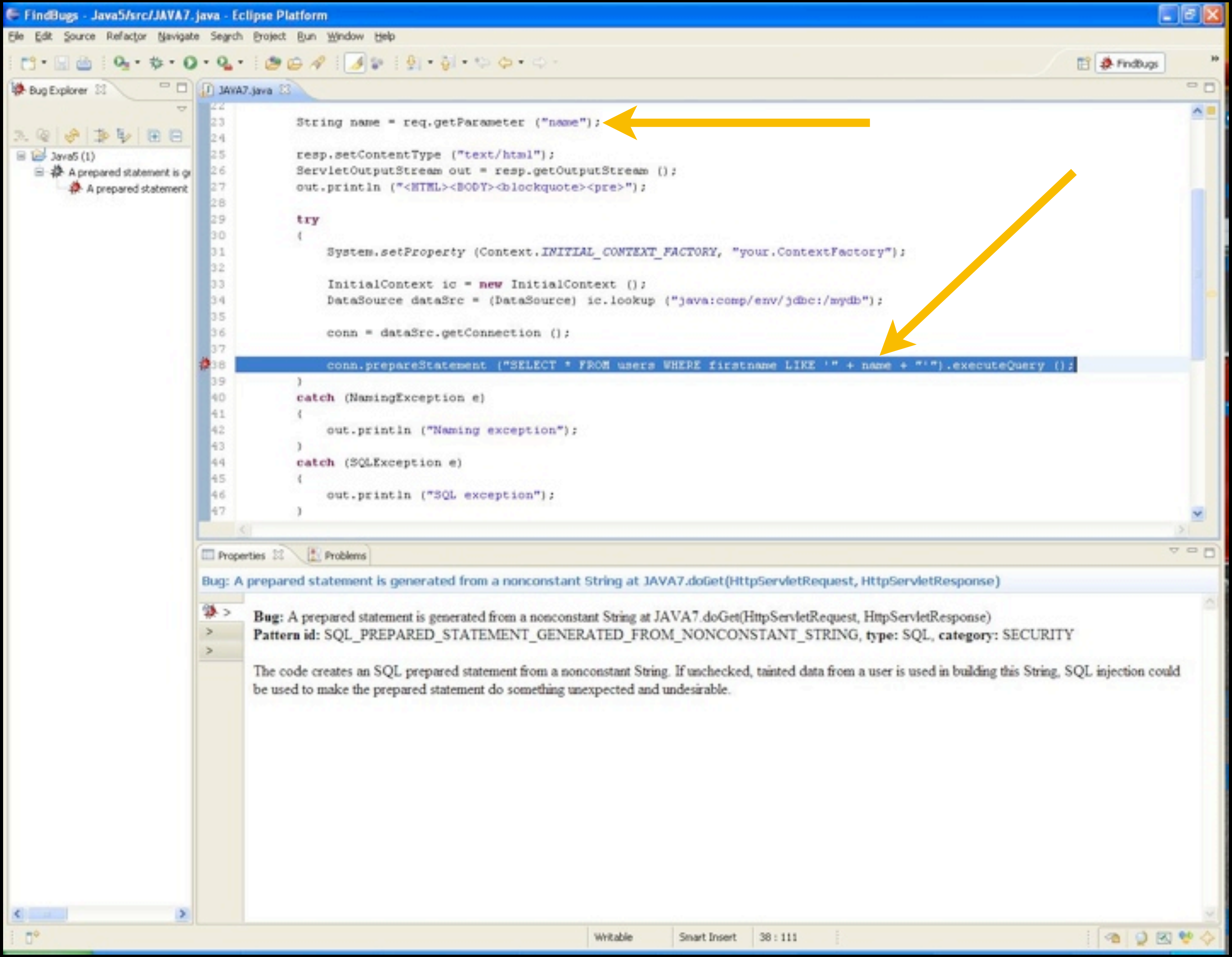
The code creates an SQL prepared statement from a nonconstant String. If unchecked, tainted data from a user is used in building this String, SQL injection could be used to make the prepared statement do something unexpected and undesirable.


```
String name = req.getParameter ("name");

resp.setContentType ("text/html");
ServletOutputStream out = resp.getOutputStream ();
out.println ("<HTML><BODY><blockquote><pre>");

try
{
    System.setProperty (Context.INITIAL_CONTEXT_FACTORY, "your.ContextFactory");

    InitialContext ic = new InitialContext ();
    DataSource dataSrc = (DataSource) ic.lookup ("java:comp/env/jdbc:/mydb");

    conn = dataSrc.getConnection ();

    conn.prepareStatement ("SELECT * FROM users WHERE firstname LIKE '" + name + "'").executeQuery ();
}
catch (NamingException e)
{
    out.println ("Naming exception");
}
catch (SQLException e)
{
    out.println ("SQL exception");
}
```

Bug: A prepared statement is generated from a nonconstant String at JAVA7.doGet(HttpServletRequest, HttpServletResponse)

Bug: A prepared statement is generated from a nonconstant String at JAVA7.doGet(HttpServletRequest, HttpServletResponse)
Pattern id: SQL_PREPARED_STATEMENT_GENERATED_FROM_NONCONSTANT_STRING, **type:** SQL, **category:** SECURITY

The code creates an SQL prepared statement from a nonconstant String. If unchecked, tainted data from a user is used in building this String, SQL injection could be used to make the prepared statement do something unexpected and undesirable.

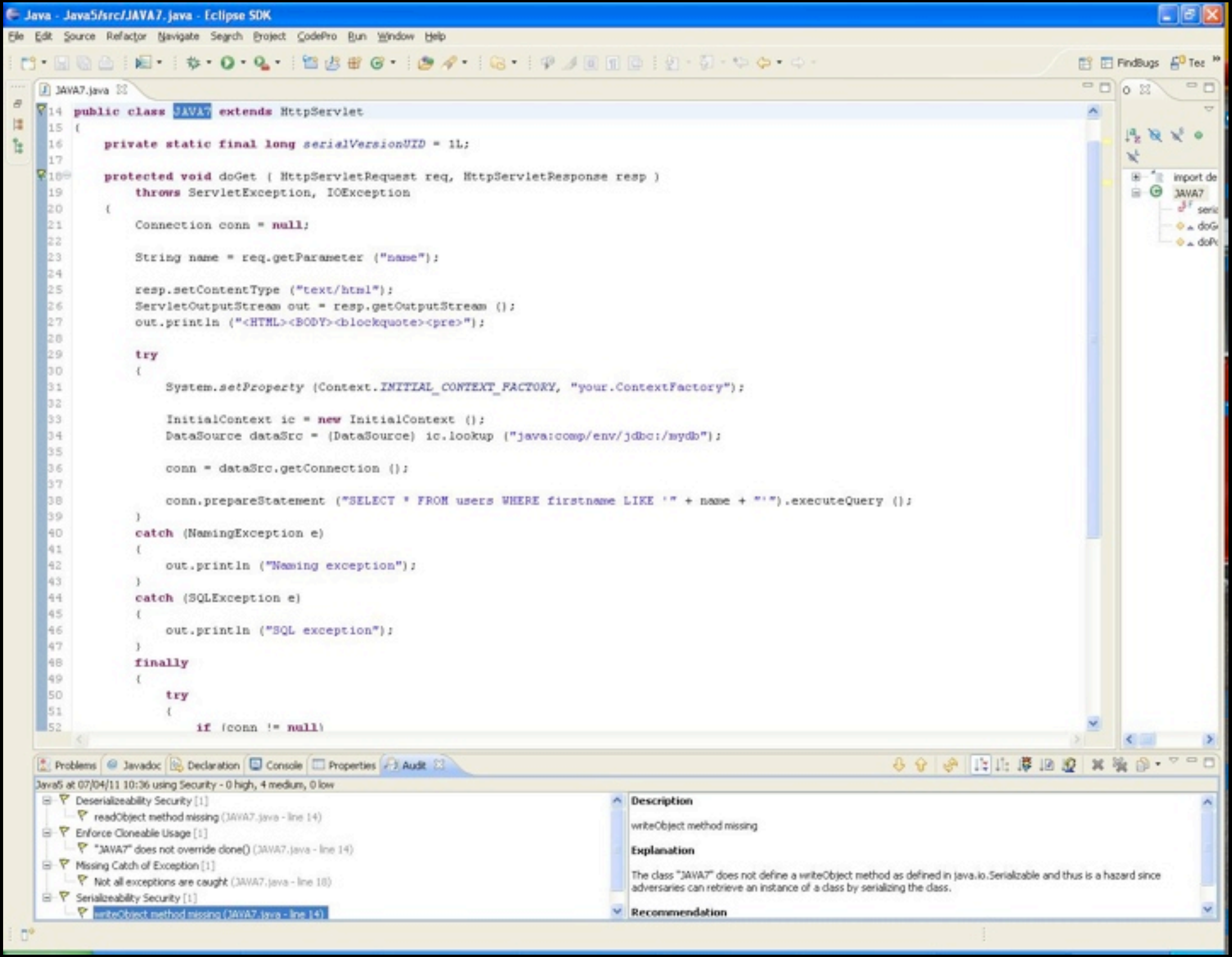

FindBugs - Java5/src/JAVA7.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
FindBugs
Bug Explorer
Java5 (1)
A prepared statement is g
A prepared statement
JAVA7.java
Properties
Problems
Writable
Smart Insert
38 : 111

```
        String name = req.getParameter ("name");

        resp.setContentType ("text/html");
        ServletOutputStream out = resp.getOutputStream ();
        out.println ("<HTML><BODY><blockquote><pre>");

        try
        {
            System.setProperty (Context.INITIAL_CONTEXT_FACTORY, "your.ContextFactory");

            InitialContext ic = new InitialContext ();
            DataSource dataSrc = (DataSource) ic.lookup ("java:comp/env/jdbc:/mydb");

            conn = dataSrc.getConnection ();

            conn.prepareStatement ("SELECT * FROM users WHERE firstname LIKE '" + name + "'").executeQuery ();
        }
        catch (NamingException e)
        {
            out.println ("Naming exception");
        }
        catch (SQLException e)
        {
            out.println ("SQL exception");
        }
```

Bug: A prepared statement is generated from a nonconstant String at JAVA7.doGet(HttpServletRequest, HttpServletResponse)

Bug: A prepared statement is generated from a nonconstant String at JAVA7.doGet(HttpServletRequest, HttpServletResponse)
Pattern id: SQL_PREPARED_STATEMENT_GENERATED_FROM_NONCONSTANT_STRING, **type:** SQL, **category:** SECURITY

The code creates an SQL prepared statement from a nonconstant String. If unchecked, tainted data from a user is used in building this String, SQL injection could be used to make the prepared statement do something unexpected and undesirable.



Java - Java5/src/JAVA7.java - Eclipse SDK

```java
public class JAVA7 extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void doGet ( HttpServletRequest req, HttpServletResponse resp )
        throws ServletException, IOException
    {
        Connection conn = null;

        String name = req.getParameter ("name");

        resp.setContentType ("text/html");
        ServletOutputStream out = resp.getOutputStream ();
        out.println ("<HTML><BODY><blockquote><pre>");

        try
        {
            System.setProperty (Context.INITIAL_CONTEXT_FACTORY, "your.ContextFactory");

            InitialContext ic = new InitialContext ();
            DataSource dataSrc = (DataSource) ic.lookup ("java:comp/env/jdbc:/mydb");

            conn = dataSrc.getConnection ();

            conn.prepareStatement ("SELECT * FROM users WHERE firstname LIKE '" + name + "'").executeQuery ();
        }
        catch (NamingException e)
        {
            out.println ("Naming exception");
        }
        catch (SQLException e)
        {
            out.println ("SQL exception");
        }
        finally
        {
            try
            {
                if (conn != null)
```

Java5 at 07/04/11 10:36 using Security - 0 high, 4 medium, 0 low

- Deserializeability Security [1]
  - readObject method missing (JAVA7.java - line 14)
- Enforce Cloneable Usage [1]
  - "JAVA7" does not override clone() (JAVA7.java - line 14)
- Missing Catch of Exception [1]
  - Not all exceptions are caught (JAVA7.java - line 18)
- Serializeability Security [1]
  - writeObject method missing (JAVA7.java - line 14)

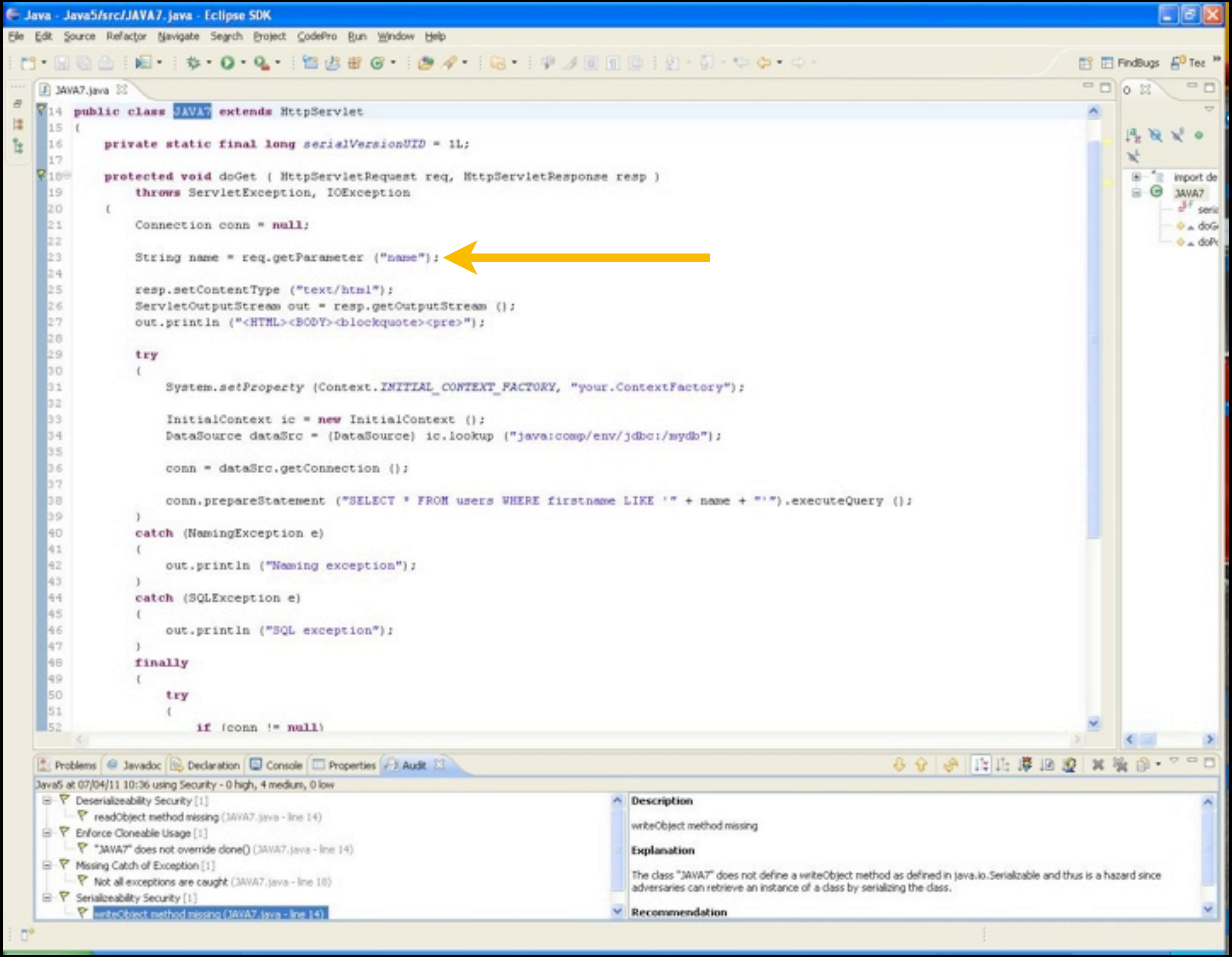
Description

writeObject method missing

Explanation

The class "JAVA7" does not define a writeObject method as defined in java.io.Serializable and thus is a hazard since adversaries can retrieve an instance of a class by serializing the class.

Recommendation



Java - Java5/src/JAVA7.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project CodePro Run Window Help
FindBugs
JAVA7.java
Problems Javadoc Declaration Console Properties Audit

```java
public class JAVA7 extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void doGet ( HttpServletRequest req, HttpServletResponse resp )
        throws ServletException, IOException
    {
        Connection conn = null;

        String name = req.getParameter ("name");

        resp.setContentType ("text/html");
        ServletOutputStream out = resp.getOutputStream ();
        out.println ("<HTML><BODY><blockquote><pre>");

        try
        {
            System.setProperty (Context.INITIAL_CONTEXT_FACTORY, "your.ContextFactory");

            InitialContext ic = new InitialContext ();
            DataSource dataSrc = (DataSource) ic.lookup ("java:comp/env/jdbc:/mydb");

            conn = dataSrc.getConnection ();

            conn.prepareStatement ("SELECT * FROM users WHERE firstname LIKE '" + name + "'").executeQuery ();
        }
        catch (NamingException e)
        {
            out.println ("Naming exception");
        }
        catch (SQLException e)
        {
            out.println ("SQL exception");
        }
        finally
        {
            try
            {
                if (conn != null)
```

```
Java5 at 07/04/11 10:36 using Security - 0 high, 4 medium, 0 low
Deserializeability Security [1]
    readObject method missing (JAVA7.java - line 14)
Enforce Cloneable Usage [1]
    "JAVA7" does not override clone() (JAVA7.java - line 14)
Missing Catch of Exception [1]
    Not all exceptions are caught (JAVA7.java - line 18)
Serializeability Security [1]
    writeObject method missing (JAVA7.java - line 14)
```

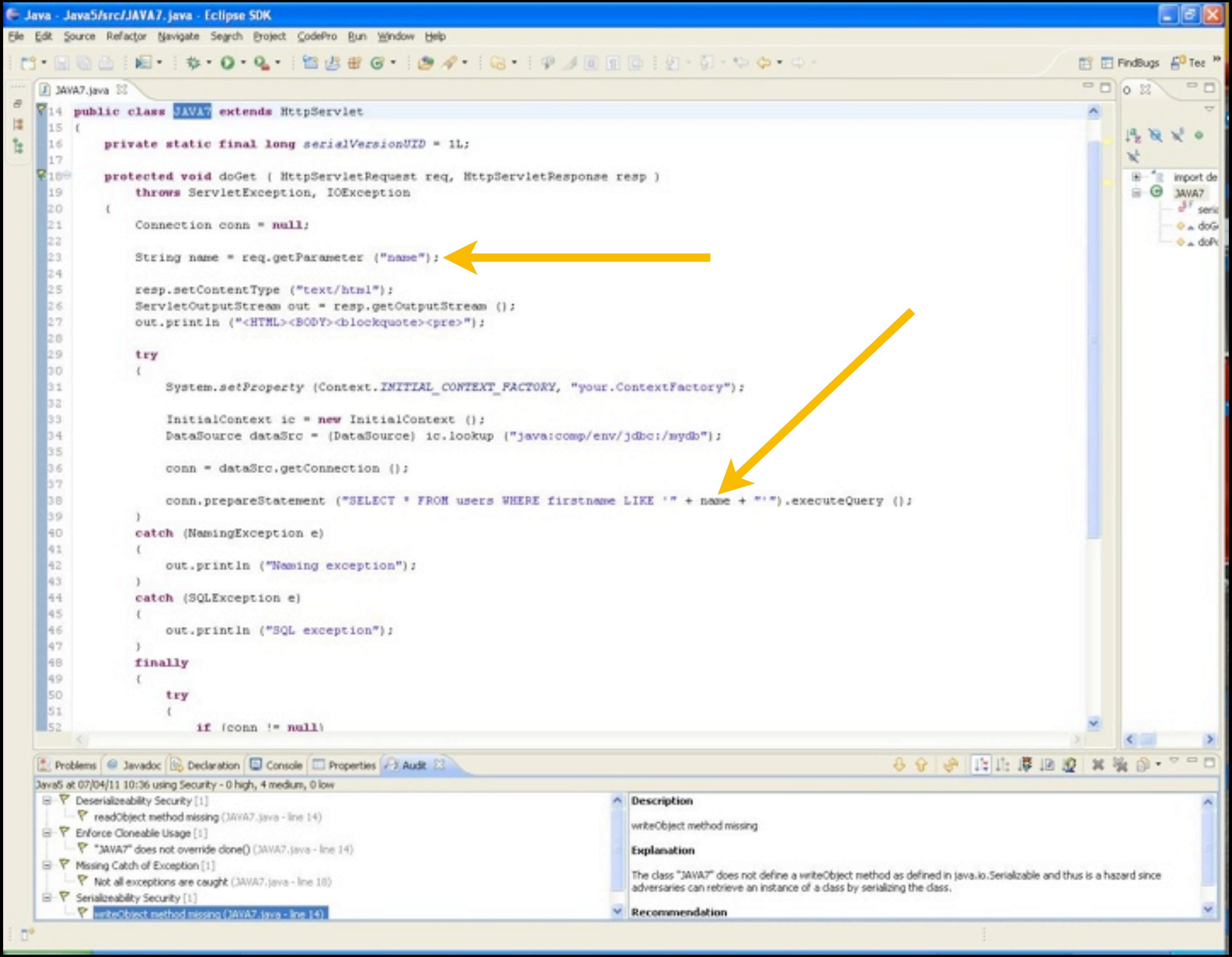
Description

writeObject method missing

Explanation

The class "JAVA7" does not define a writeObject method as defined in java.io.Serializable and thus is a hazard since adversaries can retrieve an instance of a class by serializing the class.

Recommendation


Java - Java5/src/JAVA7.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project CodePro Run Window Help
FindBugs
JAVA7.java
import de
JAVA7
Problems Javadoc Declaration Console Properties Audit

```java
public class JAVA7 extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void doGet ( HttpServletRequest req, HttpServletResponse resp )
        throws ServletException, IOException
    {
        Connection conn = null;

        String name = req.getParameter ("name");

        resp.setContentType ("text/html");
        ServletOutputStream out = resp.getOutputStream ();
        out.println ("<HTML><BODY><blockquote><pre>");

        try
        {
            System.setProperty (Context.INITIAL_CONTEXT_FACTORY, "your.ContextFactory");

            InitialContext ic = new InitialContext ();
            DataSource dataSrc = (DataSource) ic.lookup ("java:comp/env/jdbc:/mydb");

            conn = dataSrc.getConnection ();

            conn.prepareStatement ("SELECT * FROM users WHERE firstname LIKE '" + name + "'").executeQuery ();
        }
        catch (NamingException e)
        {
            out.println ("Naming exception");
        }
        catch (SQLException e)
        {
            out.println ("SQL exception");
        }
        finally
        {
            try
            {
                if (conn != null)
```

```
Java5 at 07/04/11 10:36 using Security - 0 high, 4 medium, 0 low
Deserializeability Security [1]
    readObject method missing (JAVA7.java - line 14)
Enforce Cloneable Usage [1]
    "JAVA7" does not override clone() (JAVA7.java - line 14)
Missing Catch of Exception [1]
    Not all exceptions are caught (JAVA7.java - line 18)
Serializeability Security [1]
    writeObject method missing (JAVA7.java - line 14)
```

Description

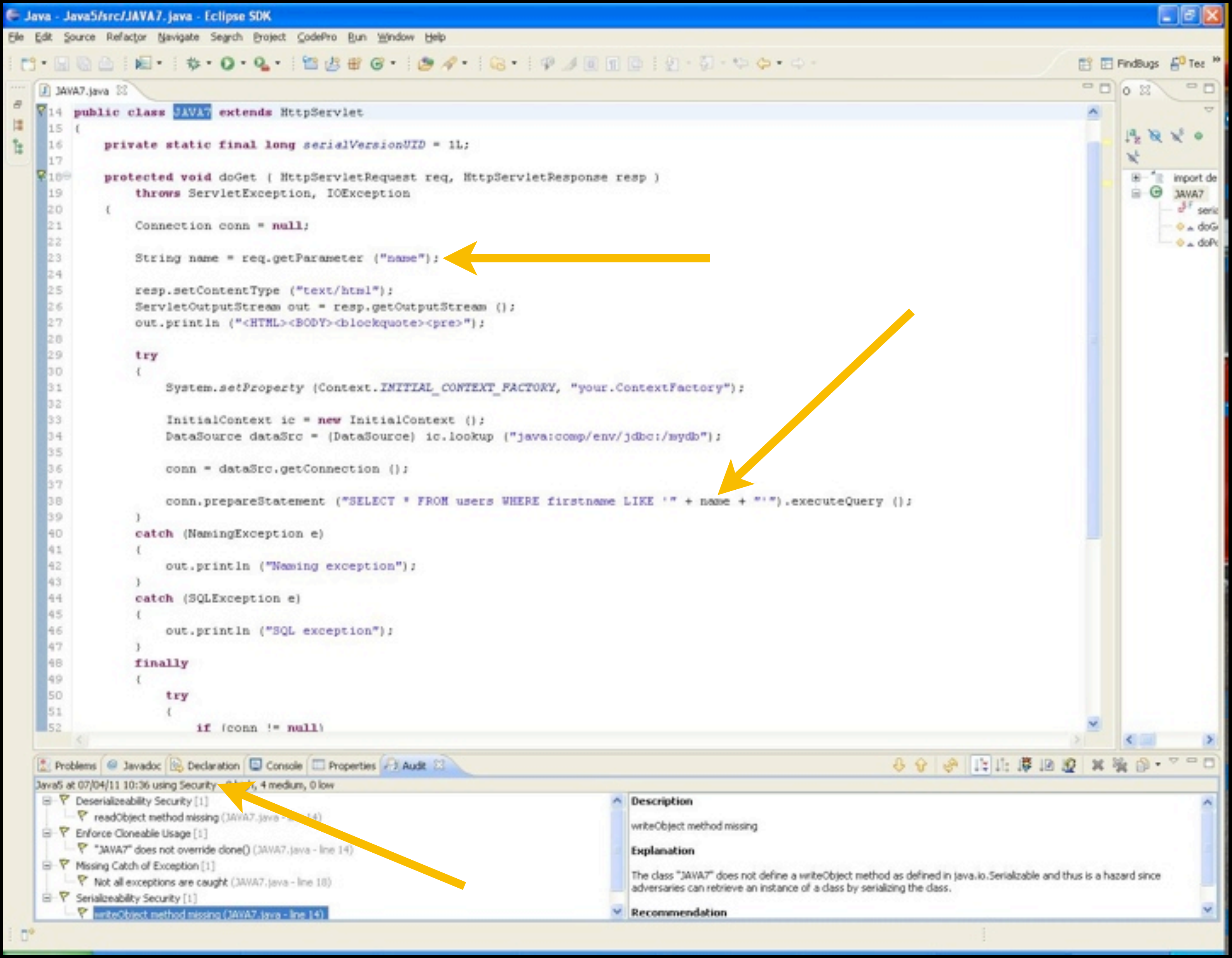
writeObject method missing

Explanation

The class "JAVA7" does not define a writeObject method as defined in java.io.Serializable and thus is a hazard since adversaries can retrieve an instance of a class by serializing the class.

Recommendation

Friday, 17 June 2011



Java - Java5/src/JAVA7.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project CodePro Run Window Help
FindBugs
JAVA7.java
Problems Javadoc Declaration Console Properties Audit

```
public class JAVA7 extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void doGet ( HttpServletRequest req, HttpServletResponse resp )
        throws ServletException, IOException
    {
        Connection conn = null;

        String name = req.getParameter ("name");

        resp.setContentType ("text/html");
        ServletOutputStream out = resp.getOutputStream ();
        out.println ("<HTML><BODY><blockquote><pre>");

        try
        {
            System.setProperty (Context.INITIAL_CONTEXT_FACTORY, "your.ContextFactory");

            InitialContext ic = new InitialContext ();
            DataSource dataSrc = (DataSource) ic.lookup ("java:comp/env/jdbc:/mydb");

            conn = dataSrc.getConnection ();

            conn.prepareStatement ("SELECT * FROM users WHERE firstname LIKE '" + name + "'").executeQuery ();
        }
        catch (NamingException e)
        {
            out.println ("Naming exception");
        }
        catch (SQLException e)
        {
            out.println ("SQL exception");
        }
        finally
        {
            try
            {
                if (conn != null)
```

```
Java5 at 07/04/11 10:36 using Security - [illegible], 4 medium, 0 low
⊟ Deserializeability Security [1]
    readObject method missing (JAVA7.java - [illegible] 14)
⊟ Enforce Cloneable Usage [1]
    "JAVA7" does not override clone() (JAVA7.java - line 14)
⊟ Missing Catch of Exception [1]
    Not all exceptions are caught (JAVA7.java - line 18)
⊟ Serializeability Security [1]
    writeObject method missing (JAVA7.java - line 14)
```

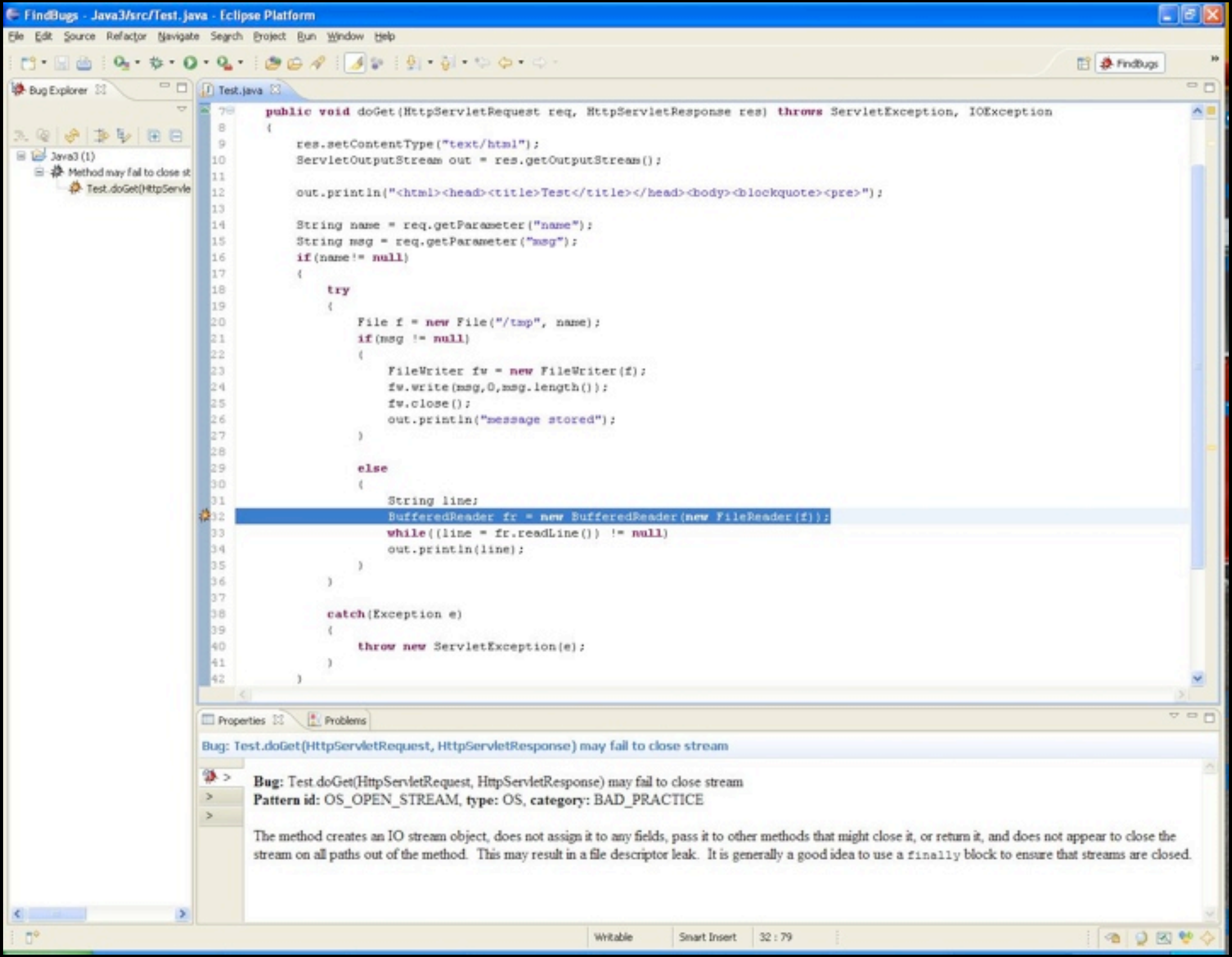
Description

writeObject method missing

Explanation

The class "JAVA7" does not define a writeObject method as defined in java.io.Serializable and thus is a hazard since adversaries can retrieve an instance of a class by serializing the class.

Recommendation

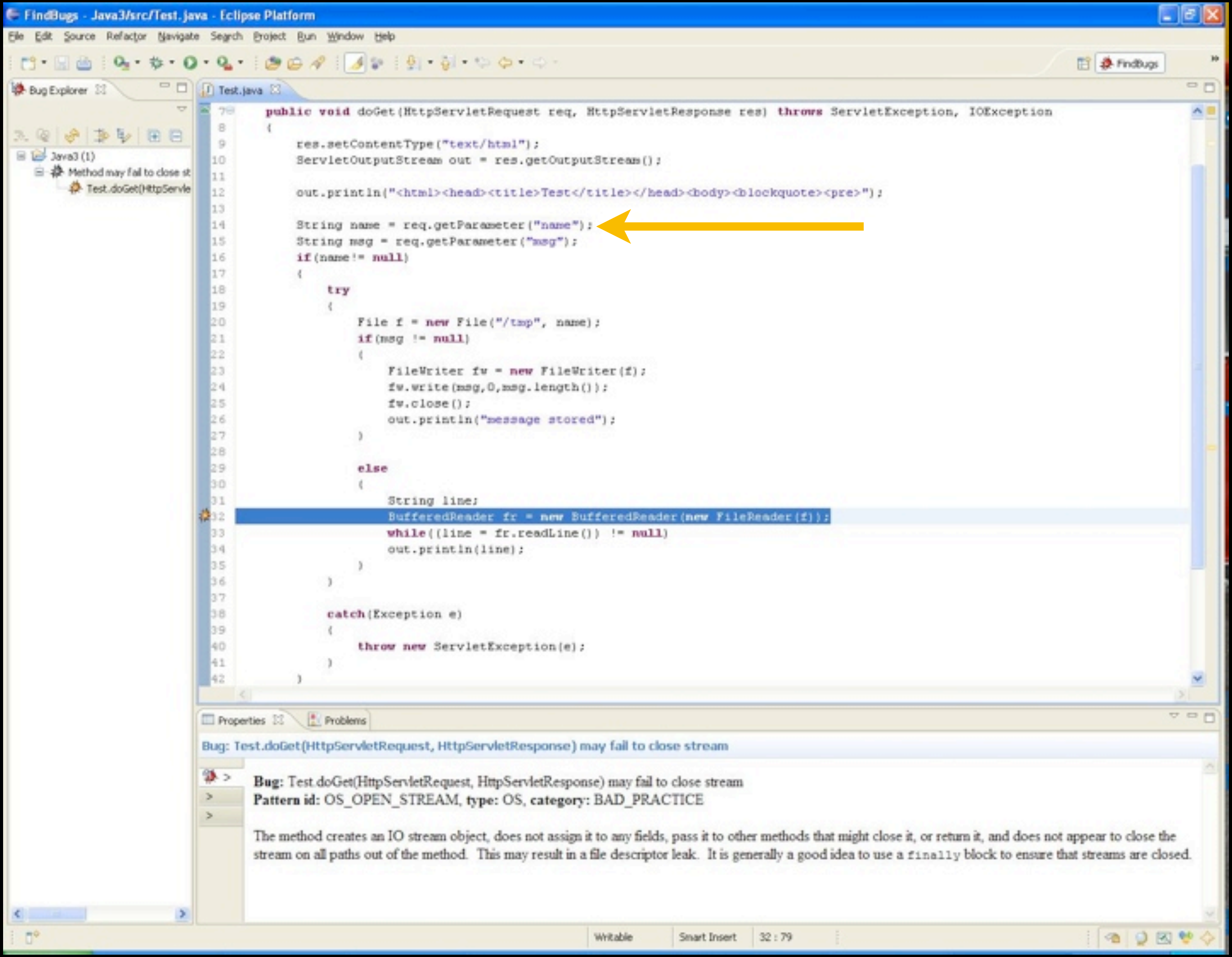

FindBugs - Java3/src/Test.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
FindBugs
Bug Explorer
Java3 (1)
Method may fail to close st
Test.doGet(HttpServle
Test.java
Properties
Problems
Writable
Smart Insert
32 : 79

```java
public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
{
    res.setContentType("text/html");
    ServletOutputStream out = res.getOutputStream();

    out.println("<html><head><title>Test</title></head><body><blockquote><pre>");

    String name = req.getParameter("name");
    String msg = req.getParameter("msg");
    if(name != null)
    {
        try
        {
            File f = new File("/tmp", name);
            if(msg != null)
            {
                FileWriter fw = new FileWriter(f);
                fw.write(msg,0,msg.length());
                fw.close();
                out.println("message stored");
            }

            else
            {
                String line;
                BufferedReader fr = new BufferedReader(new FileReader(f));
                while((line = fr.readLine()) != null)
                out.println(line);
            }
        }

        catch(Exception e)
        {
            throw new ServletException(e);
        }
    }
```

Bug: Test.doGet(HttpServletRequest, HttpServletResponse) may fail to close stream

Bug: Test.doGet(HttpServletRequest, HttpServletResponse) may fail to close stream
Pattern id: OS_OPEN_STREAM, **type:** OS, **category:** BAD_PRACTICE

The method creates an IO stream object, does not assign it to any fields, pass it to other methods that might close it, or return it, and does not appear to close the stream on all paths out of the method. This may result in a file descriptor leak. It is generally a good idea to use a `finally` block to ensure that streams are closed.


FindBugs - Java3/src/Test.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
FindBugs
Bug Explorer
Java3 (1)
Method may fail to close st
Test.doGet(HttpServle
Test.java
Properties
Problems
Writable
Smart Insert
32 : 79

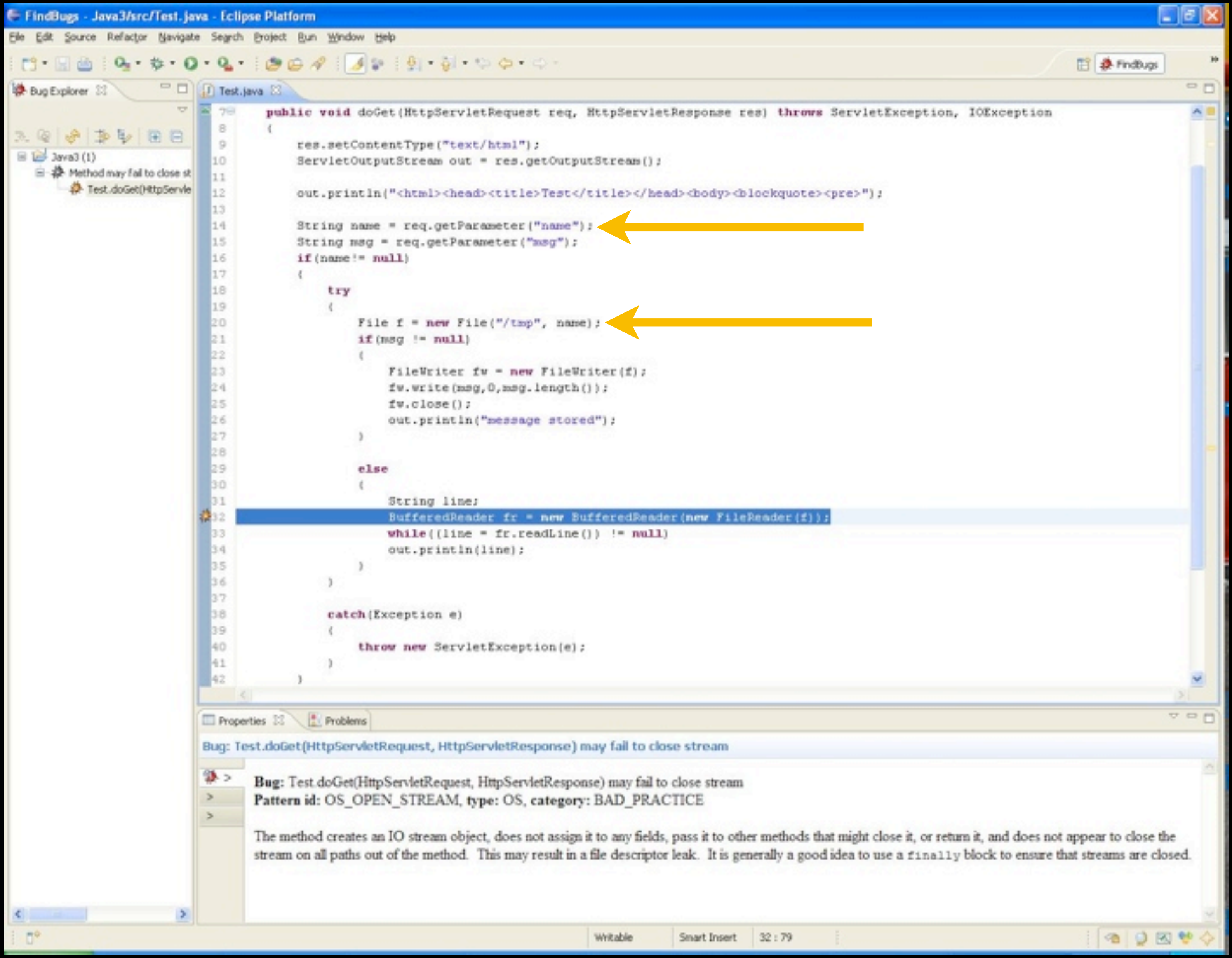
```java
public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
{
    res.setContentType("text/html");
    ServletOutputStream out = res.getOutputStream();

    out.println("<html><head><title>Test</title></head><body><blockquote><pre>");

    String name = req.getParameter("name");
    String msg = req.getParameter("msg");
    if(name != null)
    {
        try
        {
            File f = new File("/tmp", name);
            if(msg != null)
            {
                FileWriter fw = new FileWriter(f);
                fw.write(msg,0,msg.length());
                fw.close();
                out.println("message stored");
            }

            else
            {
                String line;
                BufferedReader fr = new BufferedReader(new FileReader(f));
                while((line = fr.readLine()) != null)
                out.println(line);
            }
        }

        catch(Exception e)
        {
            throw new ServletException(e);
        }
    }
```

Bug: Test.doGet(HttpServletRequest, HttpServletResponse) may fail to close stream

Bug: Test.doGet(HttpServletRequest, HttpServletResponse) may fail to close stream
Pattern id: OS_OPEN_STREAM, **type:** OS, **category:** BAD_PRACTICE

The method creates an IO stream object, does not assign it to any fields, pass it to other methods that might close it, or return it, and does not appear to close the stream on all paths out of the method. This may result in a file descriptor leak. It is generally a good idea to use a `finally` block to ensure that streams are closed.

Friday, 17 June 2011

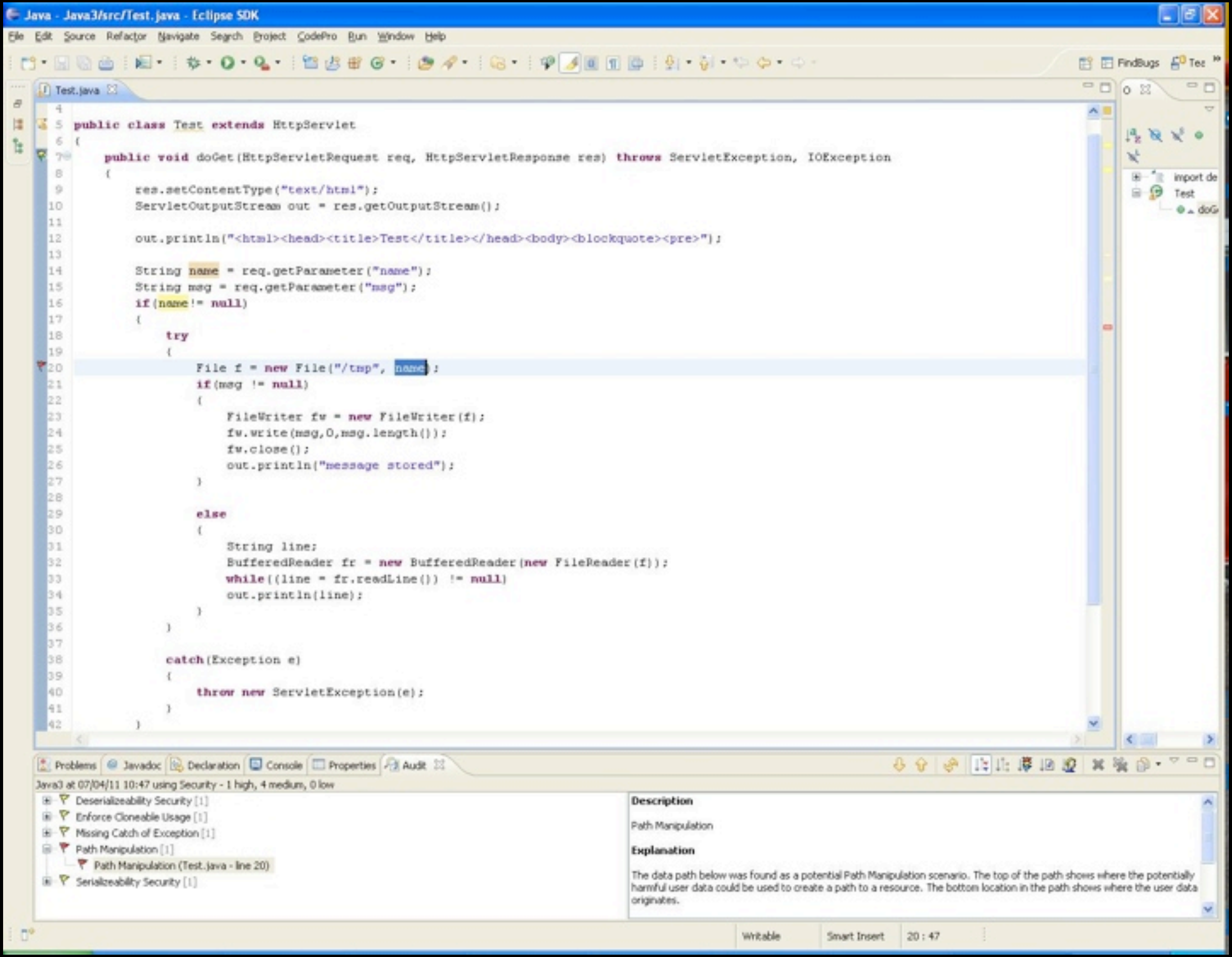

FindBugs - Java3/src/Test.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
FindBugs
Bug Explorer
Java3 (1)
Method may fail to close st
Test.doGet(HttpServle
Test.java
Properties
Problems
Writable
Smart Insert
32 : 79

```java
public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
{
    res.setContentType("text/html");
    ServletOutputStream out = res.getOutputStream();

    out.println("<html><head><title>Test</title></head><body><blockquote><pre>");

    String name = req.getParameter("name");
    String msg = req.getParameter("msg");
    if(name != null)
    {
        try
        {
            File f = new File("/tmp", name);
            if(msg != null)
            {
                FileWriter fw = new FileWriter(f);
                fw.write(msg,0,msg.length());
                fw.close();
                out.println("message stored");
            }

            else
            {
                String line;
                BufferedReader fr = new BufferedReader(new FileReader(f));
                while((line = fr.readLine()) != null)
                out.println(line);
            }
        }

        catch(Exception e)
        {
            throw new ServletException(e);
        }
    }
```

Bug: Test.doGet(HttpServletRequest, HttpServletResponse) may fail to close stream

Bug: Test.doGet(HttpServletRequest, HttpServletResponse) may fail to close stream
Pattern id: OS_OPEN_STREAM, **type:** OS, **category:** BAD_PRACTICE

The method creates an IO stream object, does not assign it to any fields, pass it to other methods that might close it, or return it, and does not appear to close the stream on all paths out of the method. This may result in a file descriptor leak. It is generally a good idea to use a `finally` block to ensure that streams are closed.


Java - Java3/src/Test.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project CodePro Run Window Help
FindBugs
Test.java

```
 4
 5  public class Test extends HttpServlet
 6  {
 7      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
 8      {
 9          res.setContentType("text/html");
10          ServletOutputStream out = res.getOutputStream();
11
12          out.println("<html><head><title>Test</title></head><body><blockquote><pre>");
13
14          String name = req.getParameter("name");
15          String msg = req.getParameter("msg");
16          if(name != null)
17          {
18              try
19              {
20                  File f = new File("/tmp", name);
21                  if(msg != null)
22                  {
23                      FileWriter fw = new FileWriter(f);
24                      fw.write(msg,0,msg.length());
25                      fw.close();
26                      out.println("message stored");
27                  }
28
29                  else
30                  {
31                      String line;
32                      BufferedReader fr = new BufferedReader(new FileReader(f));
33                      while((line = fr.readLine()) != null)
34                      out.println(line);
35                  }
36              }
37
38              catch(Exception e)
39              {
40                  throw new ServletException(e);
41              }
42          }
```

Java3 at 07/04/11 10:47 using Security - 1 high, 4 medium, 0 low

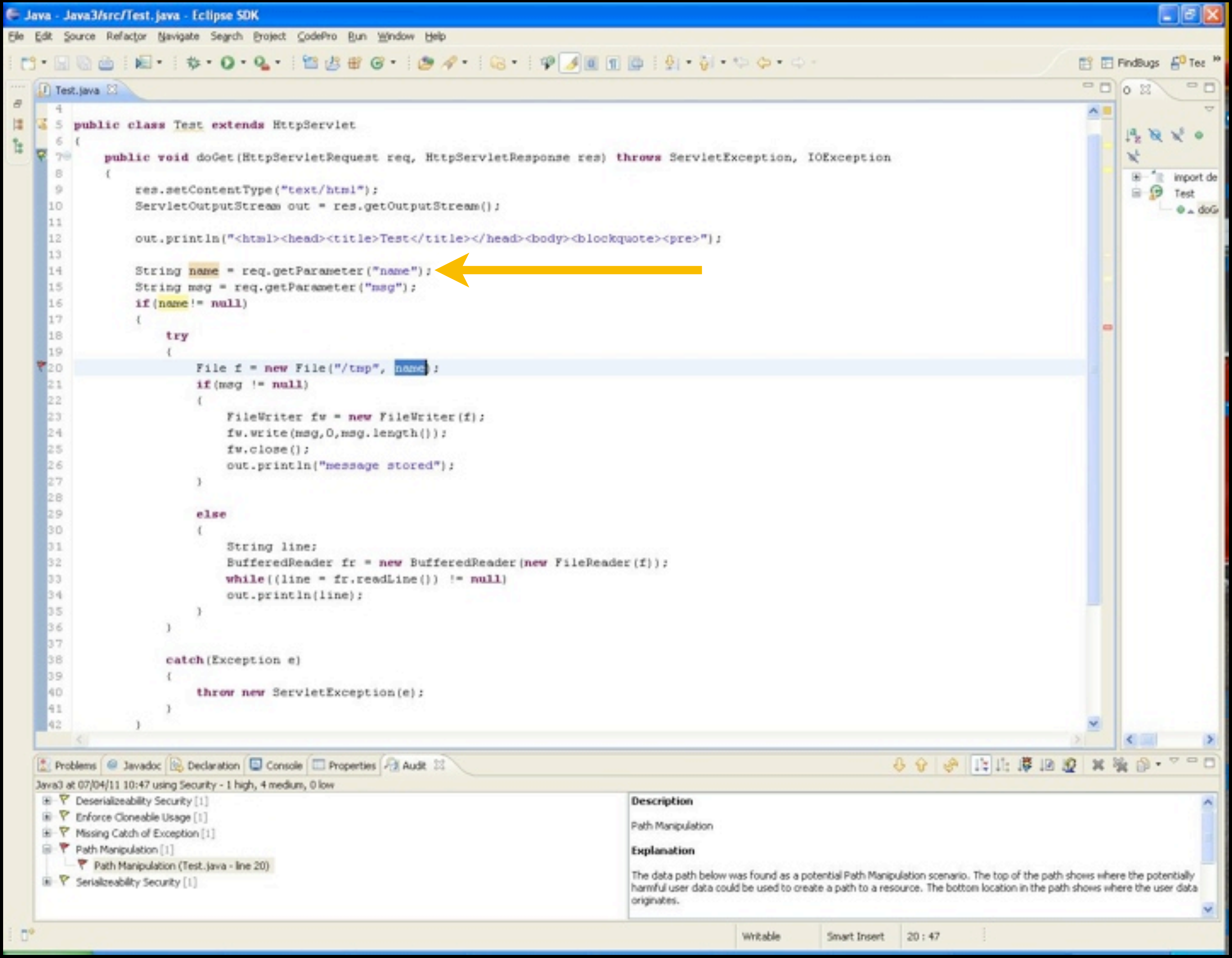
- Deserializeability Security [1]
- Enforce Cloneable Usage [1]
- Missing Catch of Exception [1]
- Path Manipulation [1]
  - Path Manipulation (Test.java - line 20)
- Serializeability Security [1]

Description

Path Manipulation

Explanation

The data path below was found as a potential Path Manipulation scenario. The top of the path shows where the potentially harmful user data could be used to create a path to a resource. The bottom location in the path shows where the user data originates.



Java - Java3/src/Test.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project CodePro Run Window Help
FindBugs
Test.java
import de
Test
doG
Problems Javadoc Declaration Console Properties Audit

```
public class Test extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
    {
        res.setContentType("text/html");
        ServletOutputStream out = res.getOutputStream();

        out.println("<html><head><title>Test</title></head><body><blockquote><pre>");

        String name = req.getParameter("name");
        String msg = req.getParameter("msg");
        if(name != null)
        {
            try
            {
                File f = new File("/tmp", name);
                if(msg != null)
                {
                    FileWriter fw = new FileWriter(f);
                    fw.write(msg,0,msg.length());
                    fw.close();
                    out.println("message stored");
                }

                else
                {
                    String line;
                    BufferedReader fr = new BufferedReader(new FileReader(f));
                    while((line = fr.readLine()) != null)
                    out.println(line);
                }
            }

            catch(Exception e)
            {
                throw new ServletException(e);
            }
        }
```

Java3 at 07/04/11 10:47 using Security - 1 high, 4 medium, 0 low

- Deserializeability Security [1]
- Enforce Cloneable Usage [1]
- Missing Catch of Exception [1]
- Path Manipulation [1]
  - Path Manipulation (Test.java - line 20)
- Serializeability Security [1]

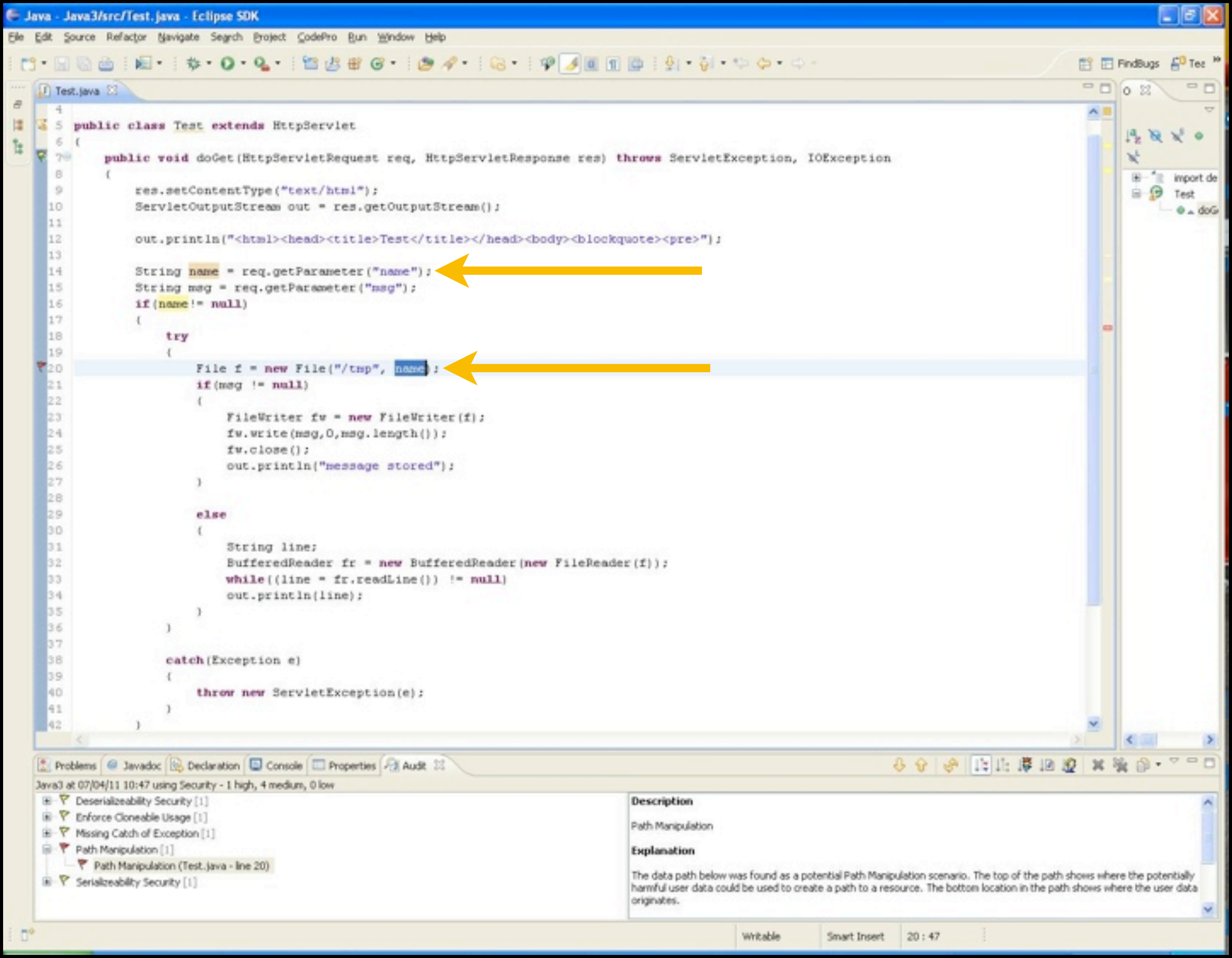
Description

Path Manipulation

Explanation

The data path below was found as a potential Path Manipulation scenario. The top of the path shows where the potentially harmful user data could be used to create a path to a resource. The bottom location in the path shows where the user data originates.

Writable Smart Insert 20 : 47



```
public class Test extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
    {
        res.setContentType("text/html");
        ServletOutputStream out = res.getOutputStream();

        out.println("<html><head><title>Test</title></head><body><blockquote><pre>");

        String name = req.getParameter("name");
        String msg = req.getParameter("msg");
        if(name != null)
        {
            try
            {
                File f = new File("/tmp", name);
                if(msg != null)
                {
                    FileWriter fw = new FileWriter(f);
                    fw.write(msg,0,msg.length());
                    fw.close();
                    out.println("message stored");
                }

                else
                {
                    String line;
                    BufferedReader fr = new BufferedReader(new FileReader(f));
                    while((line = fr.readLine()) != null)
                    out.println(line);
                }
            }

            catch(Exception e)
            {
                throw new ServletException(e);
            }
        }
```

Java3 at 07/04/11 10:47 using Security - 1 high, 4 medium, 0 low

- Deserializeability Security [1]
- Enforce Cloneable Usage [1]
- Missing Catch of Exception [1]
- Path Manipulation [1]
  - Path Manipulation (Test.java - line 20)
- Serializeability Security [1]

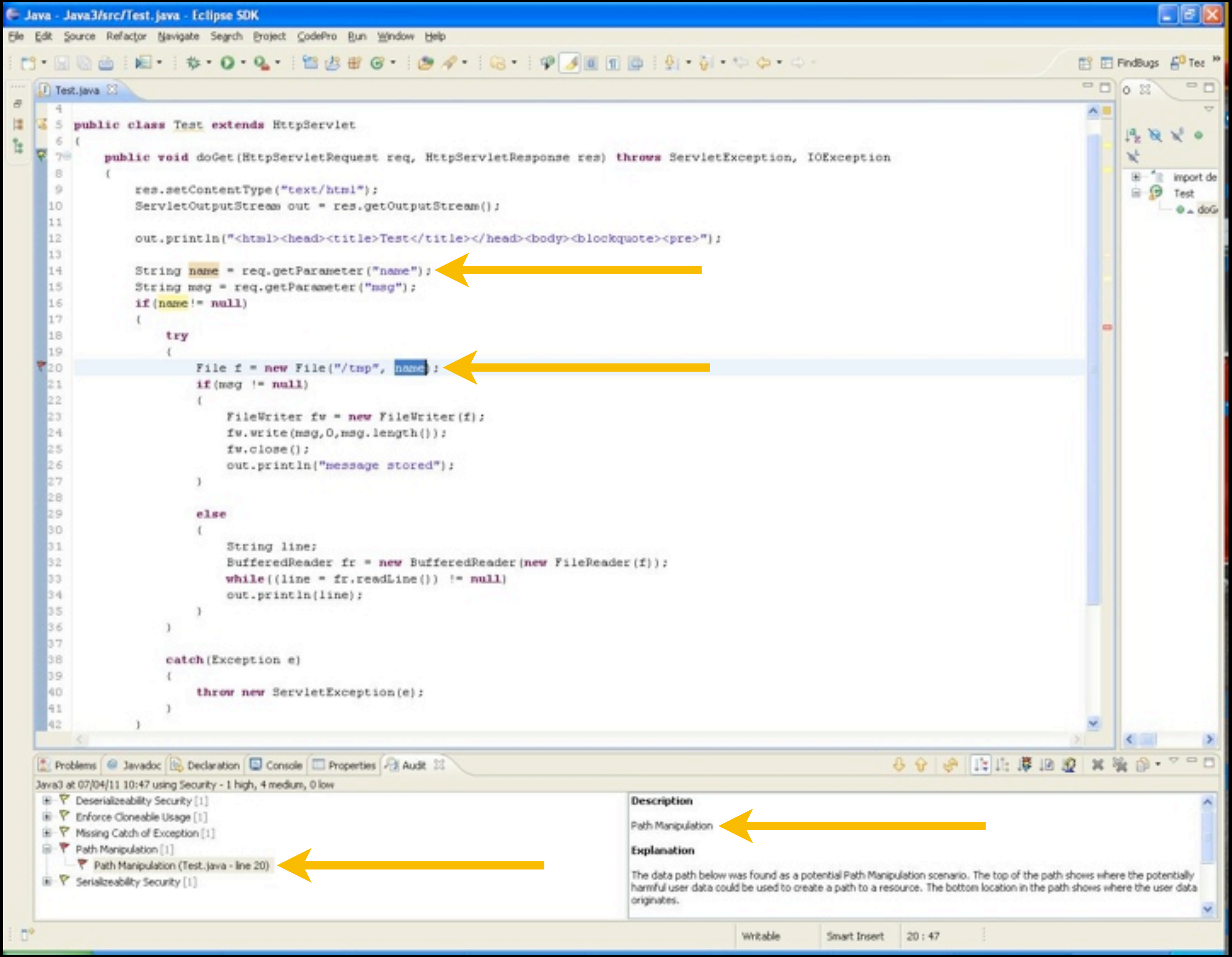
Description

Path Manipulation

Explanation

The data path below was found as a potential Path Manipulation scenario. The top of the path shows where the potentially harmful user data could be used to create a path to a resource. The bottom location in the path shows where the user data originates.

Java - Java3/src/Test.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project CodePro Run Window Help
FindBugs
Test.java
import de
Test
doG
Problems Javadoc Declaration Console Properties Audit
Writable
Smart Insert
20 : 47


Java - Java3/src/Test.java - Eclipse SDK

Test.java

```
 4
 5 public class Test extends HttpServlet
 6 {
 7     public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
 8     {
 9         res.setContentType("text/html");
10         ServletOutputStream out = res.getOutputStream();
11
12         out.println("<html><head><title>Test</title></head><body><blockquote><pre>");
13
14         String name = req.getParameter("name");
15         String msg = req.getParameter("msg");
16         if(name != null)
17         {
18             try
19             {
20                 File f = new File("/tmp", name);
21                 if(msg != null)
22                 {
23                     FileWriter fw = new FileWriter(f);
24                     fw.write(msg,0,msg.length());
25                     fw.close();
26                     out.println("message stored");
27                 }
28
29                 else
30                 {
31                     String line;
32                     BufferedReader fr = new BufferedReader(new FileReader(f));
33                     while((line = fr.readLine()) != null)
34                     out.println(line);
35                 }
36             }
37
38             catch(Exception e)
39             {
40                 throw new ServletException(e);
41             }
42         }
```

Problems | Javadoc | Declaration | Console | Properties | Audit

Java3 at 07/04/11 10:47 using Security - 1 high, 4 medium, 0 low

- Deserializeability Security [1]
- Enforce Cloneable Usage [1]
- Missing Catch of Exception [1]
- Path Manipulation [1]
  - Path Manipulation (Test.java - line 20)
- Serializeability Security [1]

Description

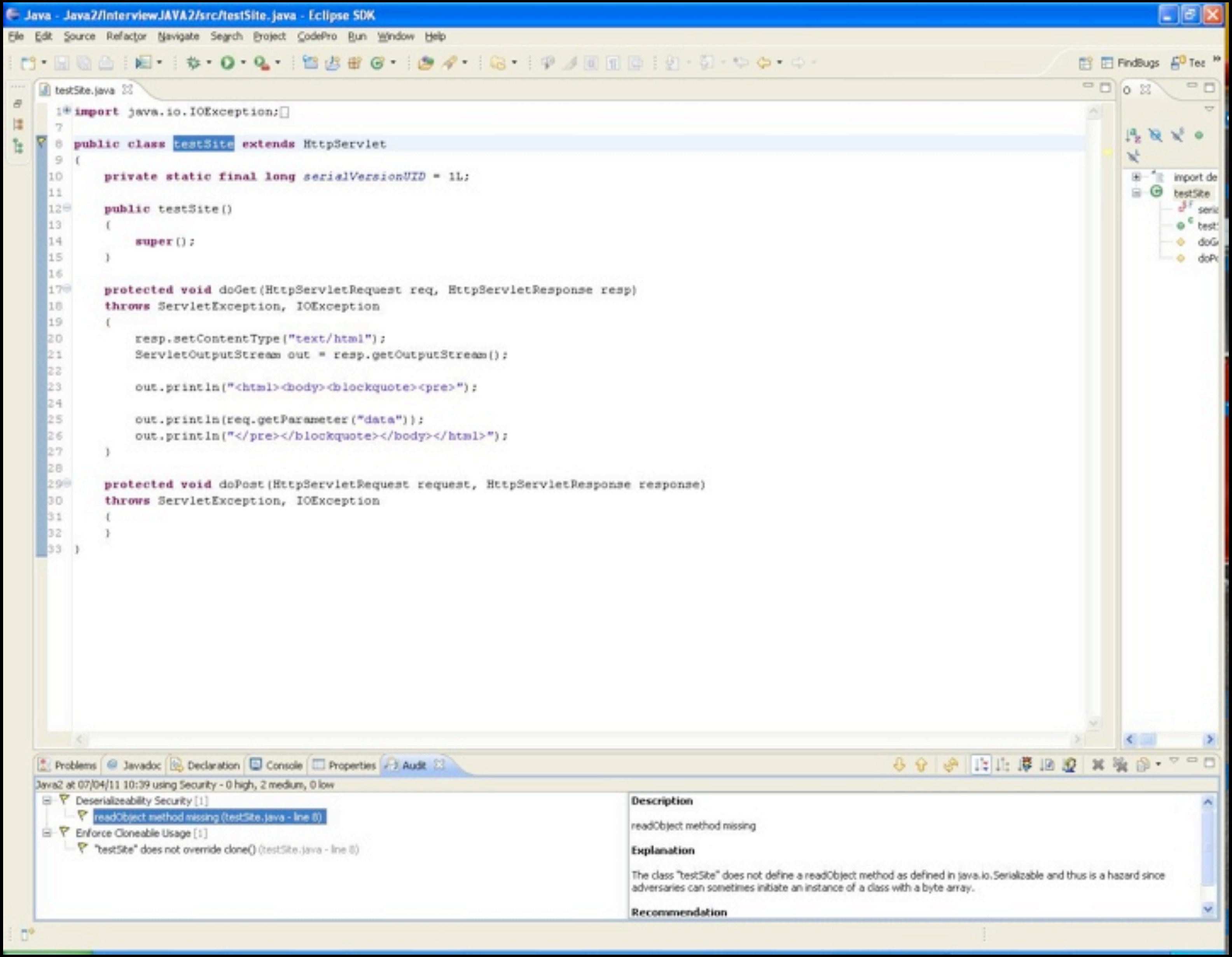
Path Manipulation

Explanation

The data path below was found as a potential Path Manipulation scenario. The top of the path shows where the potentially harmful user data could be used to create a path to a resource. The bottom location in the path shows where the user data originates.

Writable | Smart Insert | 20 : 47

Friday, 17 June 2011


Java - Java2/InterviewJAVA2/src/testSite.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project CodePro Run Window Help
FindBugs
testSite.java

```java
import java.io.IOException;

public class testSite extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    public testSite()
    {
        super();
    }

    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException
    {
        resp.setContentType("text/html");
        ServletOutputStream out = resp.getOutputStream();

        out.println("<html><body><blockquote><pre>");

        out.println(req.getParameter("data"));
        out.println("</pre></blockquote></body></html>");
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
    {
    }
}
```

```
Java2 at 07/04/11 10:39 using Security - 0 high, 2 medium, 0 low
Deserializeability Security [1]
    readObject method missing (testSite.java - line 8)
Enforce Cloneable Usage [1]
    "testSite" does not override clone() (testSite.java - line 8)
```

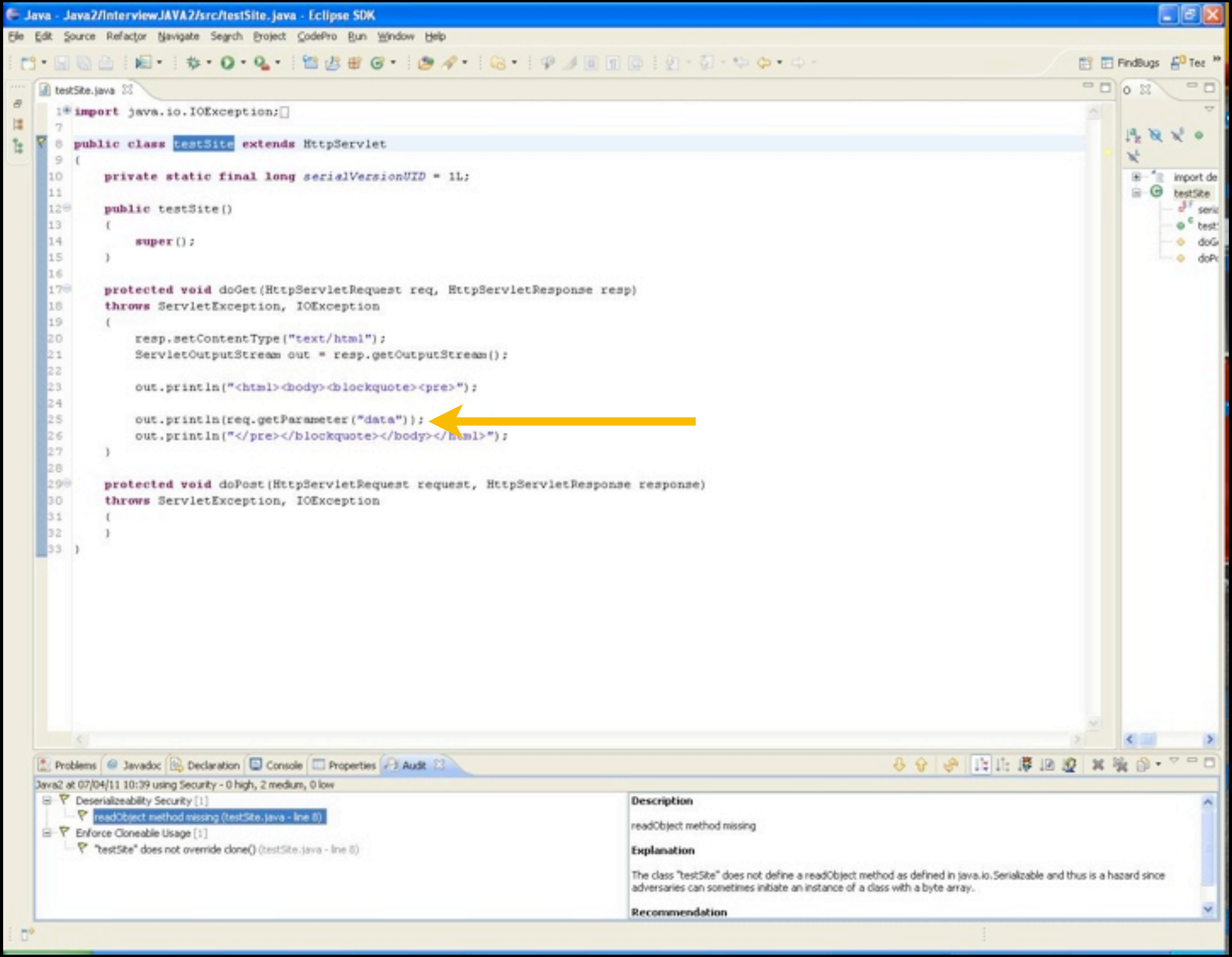
Description

readObject method missing

Explanation

The class "testSite" does not define a readObject method as defined in java.io.Serializable and thus is a hazard since adversaries can sometimes initiate an instance of a class with a byte array.

Recommendation


Java - Java2/InterviewJAVA2/src/testSite.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project CodePro Run Window Help
FindBugs
testSite.java
import de
testSite
Problems Javadoc Declaration Console Properties Audit

```
1 import java.io.IOException;
7
8 public class testSite extends HttpServlet
9 {
10     private static final long serialVersionUID = 1L;
11
12     public testSite()
13     {
14         super();
15     }
16
17     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
18     throws ServletException, IOException
19     {
20         resp.setContentType("text/html");
21         ServletOutputStream out = resp.getOutputStream();
22
23         out.println("<html><body><blockquote><pre>");
24
25         out.println(req.getParameter("data"));
26         out.println("</pre></blockquote></body></html>");
27     }
28
29     protected void doPost(HttpServletRequest request, HttpServletResponse response)
30     throws ServletException, IOException
31     {
32     }
33 }
```

Java2 at 07/04/11 10:39 using Security - 0 high, 2 medium, 0 low

- Deserializeability Security [1]
  - readObject method missing (testSite.java - line 8)
- Enforce Cloneable Usage [1]
  - "testSite" does not override clone() (testSite.java - line 8)

Description

readObject method missing

Explanation

The class "testSite" does not define a readObject method as defined in java.io.Serializable and thus is a hazard since adversaries can sometimes initiate an instance of a class with a byte array.

Recommendation

The ugly security code reviews

- "Ugly reviews" implies you do actually review code
  - An unplanned magical mystery tour at the end of the SDLC
  - Unstructured, not repeatable and heavily reliant on $C_8H_{10}N_4O_2$
  - Too late in the SDLC making findings very expensive to fix

The ugly security code reviews

- “Ugly reviews” implies you do actually review code
  - An unplanned magical mystery tour at the end of the SDLC
  - Unstructured, not repeatable and heavily reliant on $C_8H_{10}N_4O_2$
  - Too late in the SDLC making findings very expensive to fix
  - Completely manual process, no tools used during reviews
  - No audit trails, no metrics........no security?
  - Better than nothing?

The bad security code reviews

- “Bad reviews” might be fine for some companies
  - A single planned code review in your SDLC
  - Some structure, normally based on finding the OWASP top 10
  - Still too late in the SDLC making findings very expensive to fix

The bad security code reviews

- “Bad reviews” might be fine for some companies
  - A single planned code review in your SDLC
  - Some structure, normally based on finding the OWASP top 10
  - Still too late in the SDLC making findings very expensive to fix
  - Some automation, usually basic code analysis tools
  - Basic audit trails still no metrics so hard to measure “anything”
  - Better than ugly reviews, might be fine for some companies

The good security code reviews

- “Good reviews” don’t happen by accident
  - Multiple reviews defined as deliverables in your SDLC
  - Structured, repeatable process with management support
  - Reviews are exit criteria for the development and test phases

The good security code reviews

- “Good reviews” don’t happen by accident
  - Multiple reviews defined as deliverables in your SDLC
  - Structured, repeatable process with management support
  - Reviews are exit criteria for the development and test phases
  - Automation used where useful freeing up the reviewer
  - Ability to produce reports, metrics and measure improvements
  - External validation of the review process and SDLC

The principles of secure development

- What are the principles of secure development?

Philosophical Application Security

Give a man a fish and you feed him for a day, teach him to fish and you feed him for a lifetime.

Philosophical Application Security

Give a man a fish and you feed him for a day, teach him to fish and you feed him for a lifetime.

I want to apply this to secure development education:

Teach a developer about a vulnerability and he will prevent it, teach him how to develop securely and he will prevent many vulnerabilities.

realex
The real time payment exchange

The current approach

Failure to Preserve Web Page Structure
Failure to Preserve SQL Query Structure
Reliance on Untrusted Inputs in a Security Decision
Missing Encryption of Sensitive Data
Incorrect Calculation of Buffer Size
Improper Control of Filename for Include/Require Statement in PHP Program
URL Redirection to Untrusted Site
Buffer Copy without Checking Size on Input
Content Spoofing
Allocation of Resource Without Limits or Throttling
Cross Site Request Forgery
Information Leakage
Injection Flaws
Cross Site Scripting
Improper Check for Unusual or Exceptional Conditions
Insufficient Transport Layer Protection
Failure to Preserve OS Command Structure
Insufficient Authorisation
Improper Limitation of a Pathname to a Restricted Directory
Improper Access Control
Insufficient Authentication
Insecure Cryptographic Storage
Race Condition
Use of Hard-coded Credentials
Session Management
Insecure Direct Object Reference
Improper Validation of Array Index
Information Exposure Through an Error Message
Abuse of Functionality
Predictable Resource Location
Download of Code Without Integrity Check
Failure to Restrict URL Access
Unvalidated Redirects and Forwards
Buffer Access with Incorrect Length Value
Security Misconfiguration
SQL Injection
Unrestricted Upload of File with Dangerous Type
Broken Authentication
Missing Authentication for Critical Function
Integer Overflow or Wraparound
Use of a Broken or Risky Cryptographic Algorithm
Incorrect Permission Assignment for Critical Resource

realex
The real time payment exchange

The Principles of Secure Development

Secure Communications

Output Validation

Input Validation

Auditing and Logging

Authorisation

Session Management

Error Handling

Secure Resource Access

Authentication

Secure Storage

Agnitio

- What is Agnitio?
  - Tool to help with manual static analysis
  - Checklist based with reviewer & developer guidance
  - Produces audit trails & enforces integrity checks
  - Single tool for security code review reports & metrics

The real time payment exchange

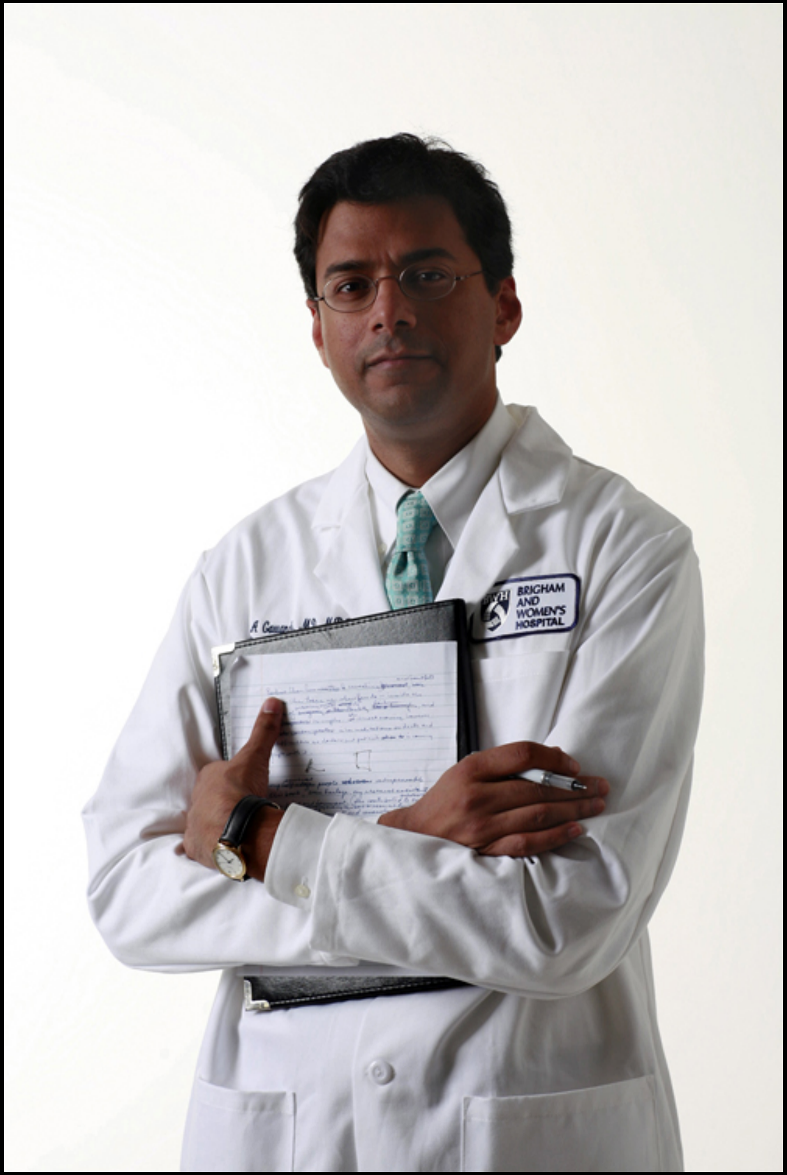
Agnitio

SECURITY

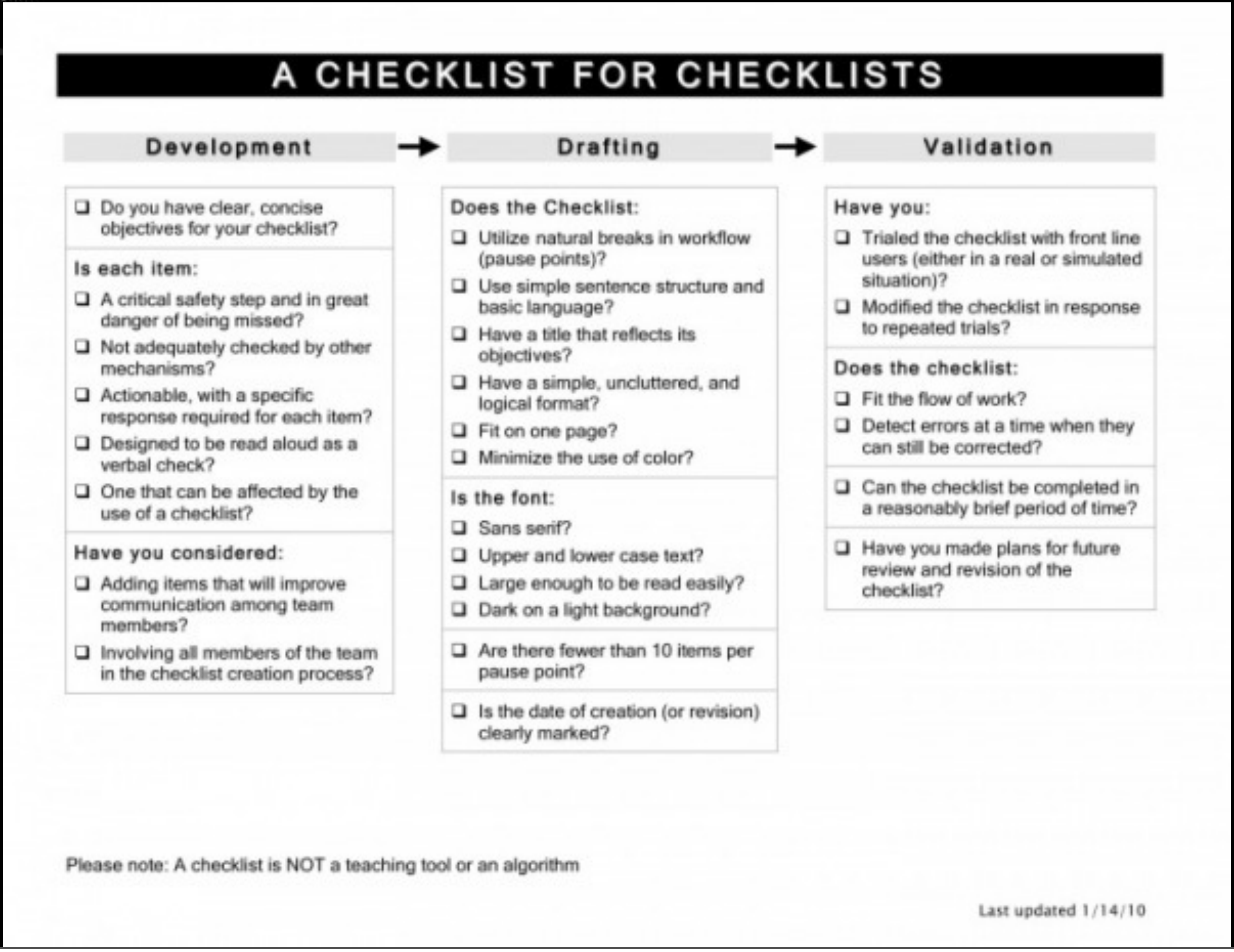
- Checklists?
  - An application for doing checklist reviews? *yawn* how boring!
  - Checklists are for n00bs! I don't need a checklist to review code!
  - I beg to differ, would you say Doctors and Pilots are n00bs?

realex
The real time payment exchange

SECURITY



realex
The real time payment exchange



A CHECKLIST FOR CHECKLISTS

Development →

- ❑ Do you have clear, concise objectives for your checklist?

Is each item:

- ❑ A critical safety step and in great danger of being missed?
- ❑ Not adequately checked by other mechanisms?
- ❑ Actionable, with a specific response required for each item?
- ❑ Designed to be read aloud as a verbal check?
- ❑ One that can be affected by the use of a checklist?

Have you considered:

- ❑ Adding items that will improve communication among team members?
- ❑ Involving all members of the team in the checklist creation process?

Drafting →

Does the Checklist:

- ❑ Utilize natural breaks in workflow (pause points)?
- ❑ Use simple sentence structure and basic language?
- ❑ Have a title that reflects its objectives?
- ❑ Have a simple, uncluttered, and logical format?
- ❑ Fit on one page?
- ❑ Minimize the use of color?

Is the font:

- ❑ Sans serif?
- ❑ Upper and lower case text?
- ❑ Large enough to be read easily?
- ❑ Dark on a light background?

- ❑ Are there fewer than 10 items per pause point?

- ❑ Is the date of creation (or revision) clearly marked?

Validation

Have you:

- ❑ Trialed the checklist with front line users (either in a real or simulated situation)?
- ❑ Modified the checklist in response to repeated trials?

Does the checklist:

- ❑ Fit the flow of work?
- ❑ Detect errors at a time when they can still be corrected?

- ❑ Can the checklist be completed in a reasonably brief period of time?

- ❑ Have you made plans for future review and revision of the checklist?

Please note: A checklist is NOT a teaching tool or an algorithm

Last updated 1/14/10

Congenital Heart Surgery Check List (Template)

Before Induction
SIGN IN

PATIENT HAS CONFIRMED

- ❑ IDENTITY
- ❑ SITE
- ❑ PROCEDURE
- ❑ CONSENT

DOES PATIENT HAVE A KNOWN ALLERGY?

- ❑ NO
- ❑ YES
  - ❑ DRUGS
  - ❑ LATEX
  - ❑ OTHER

- ❑ **H&P CURRENT (< 30d)**
- ❑ **WEIGHT RE-CHECKED**
- ❑ **ANESTHESIA SAFETY CHECK COMPLETED (Machine and Meds)**
- ❑ **PULSE OXIMETER ON PATIENT AND FUN`CTIONING**

DIFFICULT AIRWAY/ASPIRATION RISK?

- ❑ NO
- ❑ If YES, EQUIPMENT/ASSISTANCE AVAILABLE

- ❑ **INTRAVENOUS ACCESS AND FLUIDS PLANNED**
- ❑ **WARMER (blankets and fluids) IN PLACE**
- ❑ **BLOOD BANK NOTIFIED AND BLOOD PRODUCTS AVAILABLE WHEN NEEDED**

- ❑ **SIGN** (NURSING):________________
- ❑ **SIGN** (ANESTH):________________

Before Skin Incision
TIME OUT

- ❑ **CONFIRM ALL TEAM MEMBERS HAVE INTRODUCED THEMSELVES BY NAME**
- ❑ **SURGEON, ANESTHESIA, PERFUSIONIST AND NURSE VERBALLY CONFIRM**
  - ❑ PATIENT
  - ❑ SITE
  - ❑ PROCEDURE
  - ❑ IMAGING AVAILABLE AND REVIEWED
  - ❑ TRANSESOPHAGEAL ECHO (TEE) OR OTHER ECHO
  - ❑ ANTIFIBRINOLYTICS
  - ❑ ANTIBIOTICS ADMINISTERED (within last 60 min)

  PERFUSION STRATEGY:
  - ❑ CANNULATION SITES
  - ❑ CANNULAE SIZES
  - ❑ BYPASS PRIME (blood vs prime)
  - ❑ TARGETED CORE TEMP
  - ❑ USE OR NON-USE OF DHCA, SELECTIVE CEREBRAL PERFUSION
  - ❑ ICE ON THE HEAD
  - ❑ OTHER BYPASS CONSIDERATIONS (shunts, collaterals, AR, LV venting, CARDIOPLEGIA, etc)

ANESTHESIA TEAM REVIEWS:

- ❑ ANY FURTHER PATIENT-SPECIFIC CONCERNS?

NURSING TEAM REVIEWS:

- ❑ EQUIPMENT STERILITY CONFIRMED?
- ❑ ARE THERE EQUIPMENT/PROSTHESES ISSUES OR ANY CONCERNS?

- ❑ **SIGN** (SURG):____________________

Before Patient Leaves Room
SIGN OUT

NURSE VERBALLY CONFIRMS WITH THE TEAM:

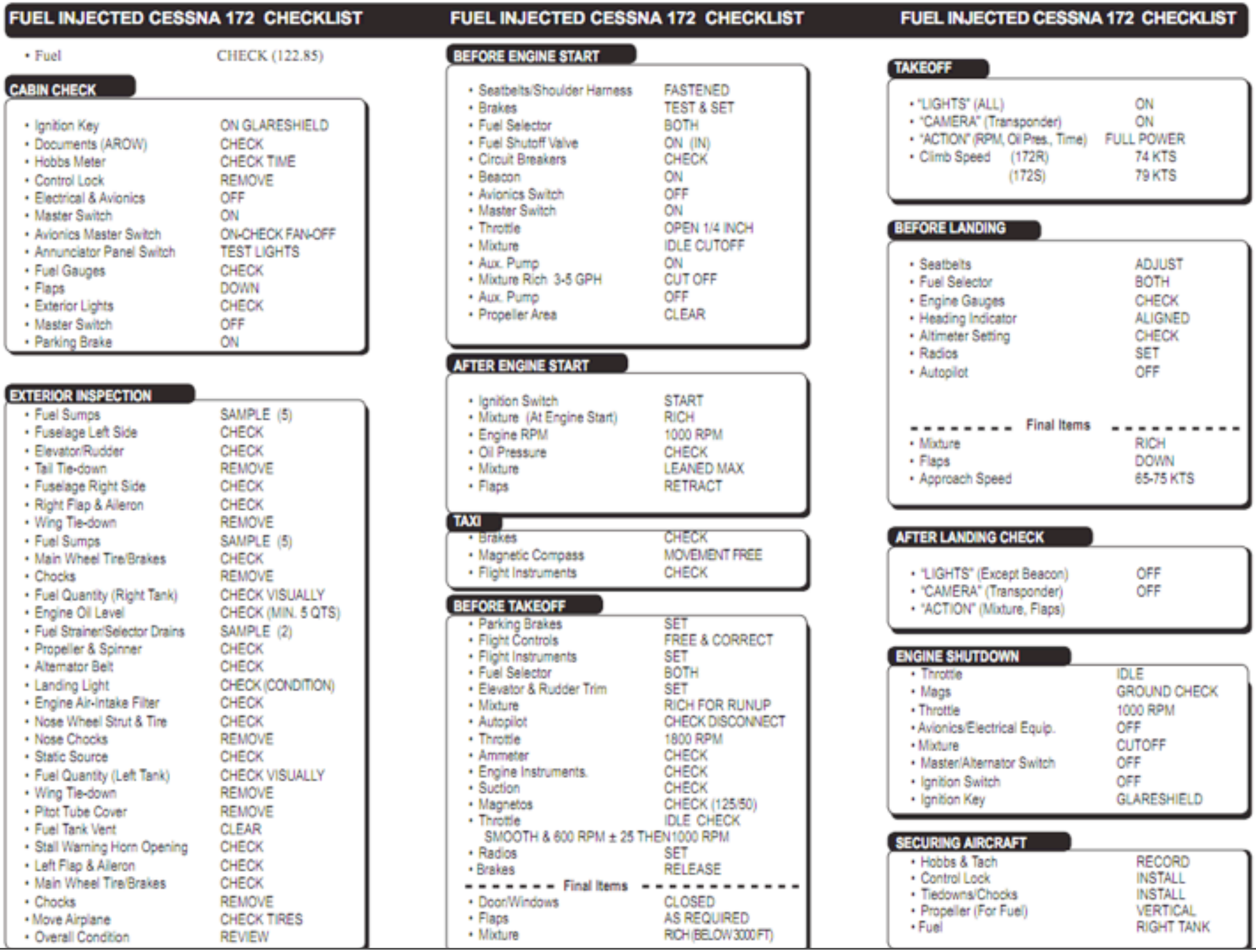
- ❑ NAME OF THE PROCEDURE
- ❑ THAT INSTRUMENT, SPONGE AND NEEDLE COUNTS ARE CORRECT

- ❑ **HOW THE SPECIMEN IS LABELLED**
  - ❑ INCLUDING PATIENT NAME
  - ❑ SENT FOR APPROPRIATE TESTS
- ❑ **WHETHER THERE ARE ANY EQUIPMENT PROBLEMS TO BE ADDRESSED**

SURGEON, ANESTHESIA PROFESSIONAL AND NURSE

- ❑ REVIEW THE KEY CONCERNS FOR POST-OP RECOVERY AND MANAGEMENT OF THIS PATIENT
- ❑ BLOOD PRODUCTS USED
- ❑ BLOOD PRODUCTS STILL AVAILABLE
- ❑ BREAKS IN TECHNIQUE

- ❑ **SIGN** (NURSING):____________________
- ❑ **SIGN** (SURG):______________________



FUEL INJECTED CESSNA 172 CHECKLIST

• Fuel	CHECK (122.85)

CABIN CHECK

• Ignition Key	ON GLARESHIELD
• Documents (AROW)	CHECK
• Hobbs Meter	CHECK TIME
• Control Lock	REMOVE
• Electrical & Avionics	OFF
• Master Switch	ON
• Avionics Master Switch	ON-CHECK FAN-OFF
• Annunciator Panel Switch	TEST LIGHTS
• Fuel Gauges	CHECK
• Flaps	DOWN
• Exterior Lights	CHECK
• Master Switch	OFF
• Parking Brake	ON

EXTERIOR INSPECTION

• Fuel Sumps	SAMPLE (5)
• Fuselage Left Side	CHECK
• Elevator/Rudder	CHECK
• Tail Tie-down	REMOVE
• Fuselage Right Side	CHECK
• Right Flap & Aileron	CHECK
• Wing Tie-down	REMOVE
• Fuel Sumps	SAMPLE (5)
• Main Wheel Tire/Brakes	CHECK
• Chocks	REMOVE
• Fuel Quantity (Right Tank)	CHECK VISUALLY
• Engine Oil Level	CHECK (MIN. 5 QTS)
• Fuel Strainer/Selector Drains	SAMPLE (2)
• Propeller & Spinner	CHECK
• Alternator Belt	CHECK
• Landing Light	CHECK (CONDITION)
• Engine Air-Intake Filter	CHECK
• Nose Wheel Strut & Tire	CHECK
• Nose Chocks	REMOVE
• Static Source	CHECK
• Fuel Quantity (Left Tank)	CHECK VISUALLY
• Wing Tie-down	REMOVE
• Pitot Tube Cover	REMOVE
• Fuel Tank Vent	CLEAR
• Stall Warning Horn Opening	CHECK
• Left Flap & Aileron	CHECK
• Main Wheel Tire/Brakes	CHECK
• Chocks	REMOVE
• Move Airplane	CHECK TIRES
• Overall Condition	REVIEW

FUEL INJECTED CESSNA 172 CHECKLIST

BEFORE ENGINE START

• Seatbelts/Shoulder Harness	FASTENED
• Brakes	TEST & SET
• Fuel Selector	BOTH
• Fuel Shutoff Valve	ON (IN)
• Circuit Breakers	CHECK
• Beacon	ON
• Avionics Switch	OFF
• Master Switch	ON
• Throttle	OPEN 1/4 INCH
• Mixture	IDLE CUTOFF
• Aux. Pump	ON
• Mixture Rich 3-5 GPH	CUT OFF
• Aux. Pump	OFF
• Propeller Area	CLEAR

AFTER ENGINE START

• Ignition Switch	START
• Mixture (At Engine Start)	RICH
• Engine RPM	1000 RPM
• Oil Pressure	CHECK
• Mixture	LEANED MAX
• Flaps	RETRACT

TAXI

• Brakes	CHECK
• Magnetic Compass	MOVEMENT FREE
• Flight Instruments	CHECK

BEFORE TAKEOFF

• Parking Brakes	SET
• Flight Controls	FREE & CORRECT
• Flight Instruments	SET
• Fuel Selector	BOTH
• Elevator & Rudder Trim	SET
• Mixture	RICH FOR RUNUP
• Autopilot	CHECK DISCONNECT
• Throttle	1800 RPM
• Ammeter	CHECK
• Engine Instruments.	CHECK
• Suction	CHECK
• Magnetos	CHECK (125/50)
• Throttle	IDLE CHECK
SMOOTH & 600 RPM ± 25 THEN	1000 RPM
• Radios	SET
• Brakes	RELEASE

- - - - - - - Final Items - - - - - - - - - - - -

• Door/Windows	CLOSED
• Flaps	AS REQUIRED
• Mixture	RICH (BELOW 3000 FT)

FUEL INJECTED CESSNA 172 CHECKLIST

TAKEOFF

• "LIGHTS" (ALL)		ON
• "CAMERA" (Transponder)		ON
• "ACTION" (RPM, Oil Pres., Time)		FULL POWER
• Climb Speed	(172R)	74 KTS
	(172S)	79 KTS

BEFORE LANDING

• Seatbelts	ADJUST
• Fuel Selector	BOTH
• Engine Gauges	CHECK
• Heading Indicator	ALIGNED
• Altimeter Setting	CHECK
• Radios	SET
• Autopilot	OFF

- - - - - - - - Final Items - - - - - - - - - -

• Mixture	RICH
• Flaps	DOWN
• Approach Speed	65-75 KTS

AFTER LANDING CHECK

• "LIGHTS" (Except Beacon)	OFF
• "CAMERA" (Transponder)	OFF
• "ACTION" (Mixture, Flaps)	

ENGINE SHUTDOWN

• Throttle	IDLE
• Mags	GROUND CHECK
• Throttle	1000 RPM
• Avionics/Electrical Equip.	OFF
• Mixture	CUTOFF
• Master/Alternator Switch	OFF
• Ignition Switch	OFF
• Ignition Key	GLARESHIELD

SECURING AIRCRAFT

• Hobbs & Tach	RECORD
• Control Lock	INSTALL
• Tiedowns/Chocks	INSTALL
• Propeller (For Fuel)	VERTICAL
• Fuel	RIGHT TANK

Friday, 17 June 2011

Agnitio

- Checklists?
  - So you don't use a checklist for reviewing source code?
  - What's the worst that could happen?

The real time payment exchange

SECURITY

Ariane 5 flight 501

esa
ariane
V 157
© 2002 - ESA - CNES - ARIANESPACE / photo service Optique Video CSG

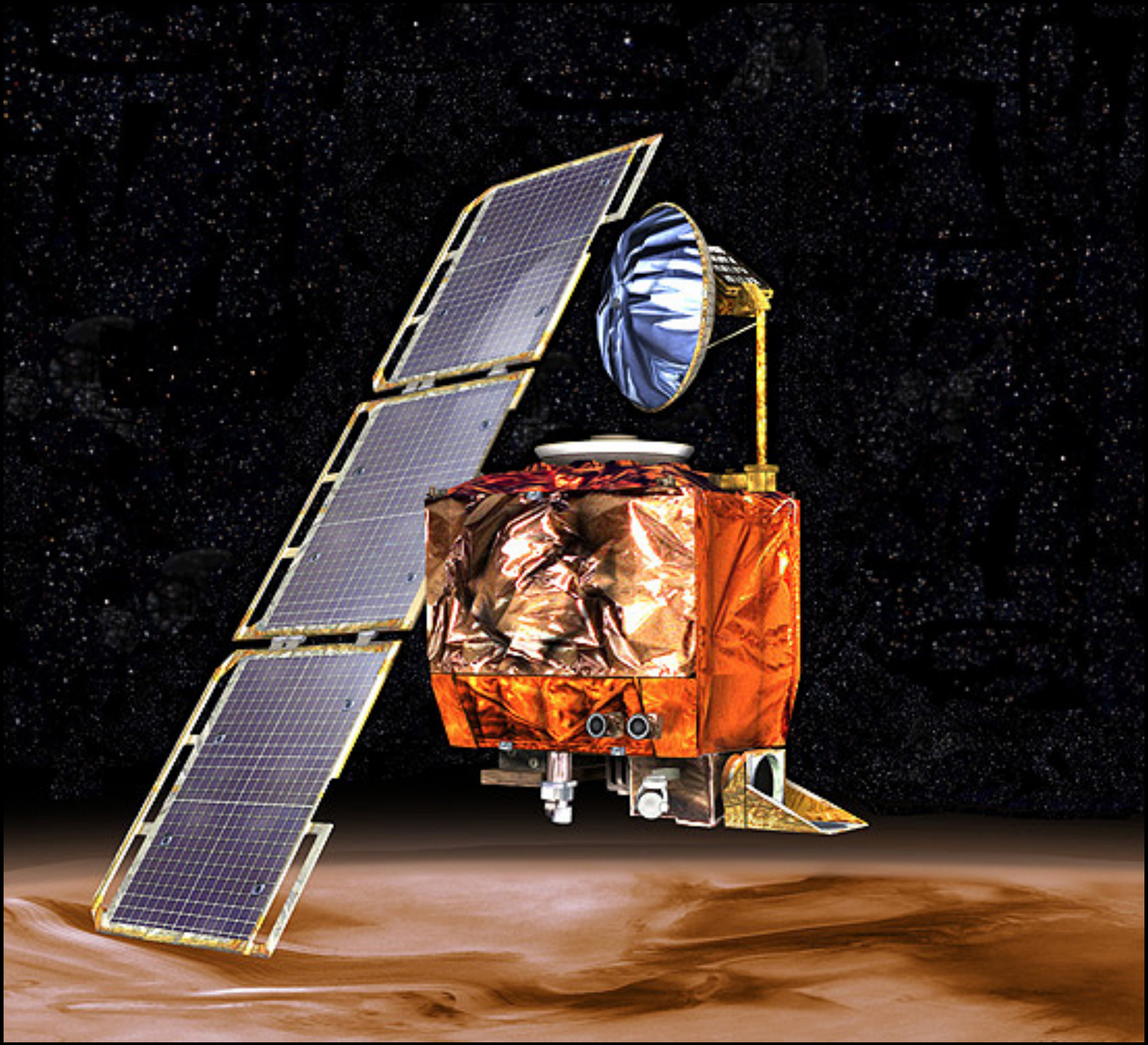
realex
The real time payment exchange

Ariane 5 flight 501

SECURITY

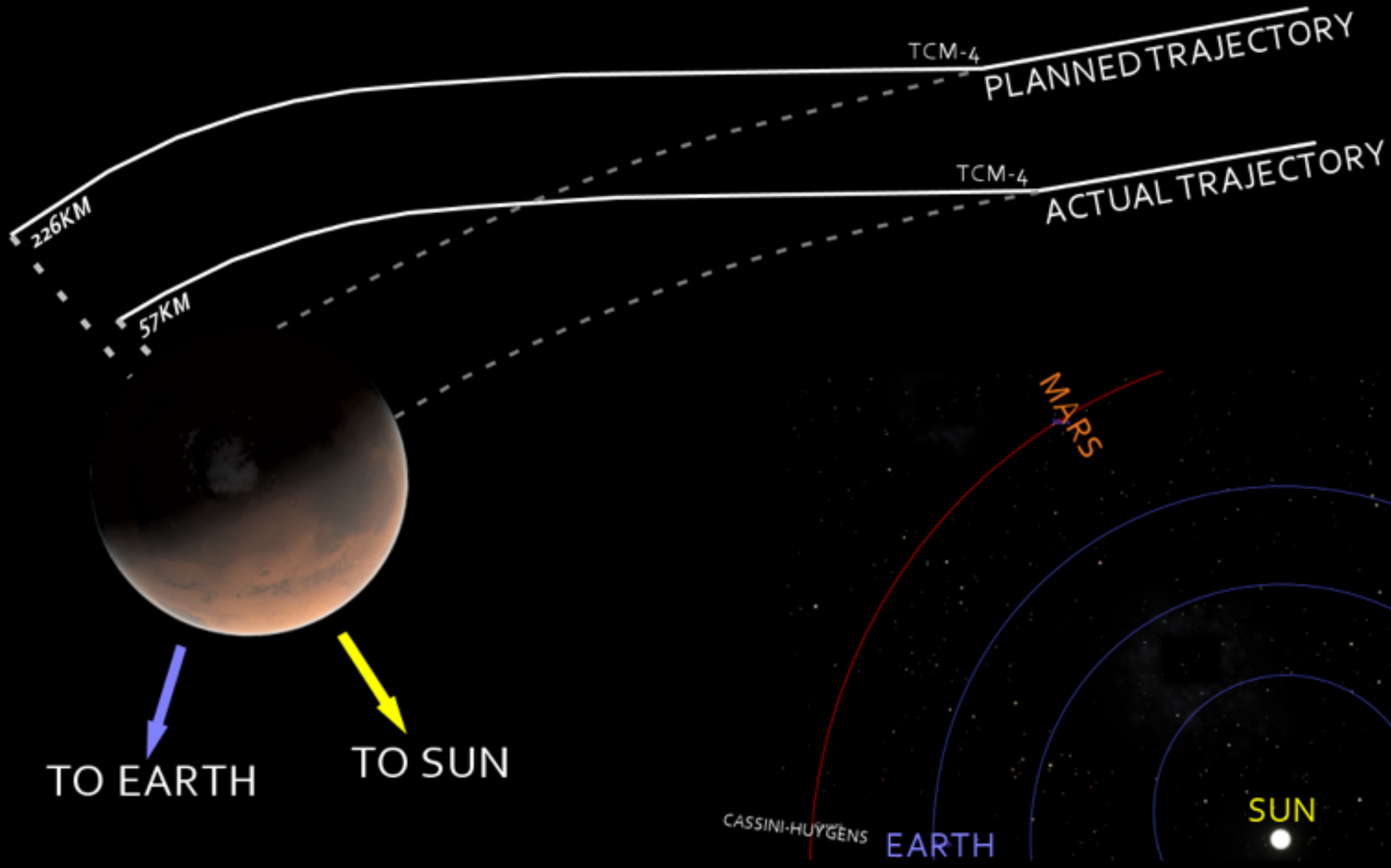
Therac-25

Mars Climate Orbiter



SECURITY

Mars Climate Orbiter



Agnitio

- Checklists?
  - So you don't use a checklist for reviewing source code?
  - What's the worst that could happen?
  - Four people dead and over €700m of equipment destroyed
  - Checklists can be useful to pilots, doctors and code reviewers!

The real time payment exchange

Agnitio

- So, why did I develop Agnitio?
  - I love using checklists for security code reviews!

Agnitio

- So, why did I develop Agnitio?
  - I love using checklists for security code reviews!
  - Even if your process is good it might not be smart

Agnitio

- So, why did I develop Agnitio?
  - I love using checklists for security code reviews!
  - Even if your process is good it might not be smart
  - Is your review process really repeatable and easy to audit?

Agnitio

- So, why did I develop Agnitio?
  - I love using checklists for security code reviews!
  - Even if your process is good it might not be smart
  - Is your review process really repeatable and easy to audit?
  - How about producing metrics, useful reports & integrity checks?

Agnitio

- So, why did I develop Agnitio?
  - I love using checklists for security code reviews!
  - Even if your process is good it might not be smart
  - Is your review process really repeatable and easy to audit?
  - How about producing metrics, useful reports & integrity checks?
  - No? That’s why I developed Agnitio!

realex
The real time payment exchange

Agnitio

- Why did I develop Agnitio?
  - My own review process was good but it wasn't smart
  - Minimum of 2 code reviews per release
  - Three pieces of evidence produced per review
  - One central Excel sheet for metrics and "audit" trail

The real time payment exchange

Why did I develop Agnitio?

- 2 reviews: 3 deliverables x ~200 releases in 2010
- 400 security code reviews

x 10

The real time payment exchange

Why did I develop Agnitio?

- 2 reviews: 3 deliverables x ~200 releases in 2010
- 400 security code reviews

Why did I develop Agnitio?

- Demonstration: security code reviews

SECURITY

Why did I develop Agnitio?

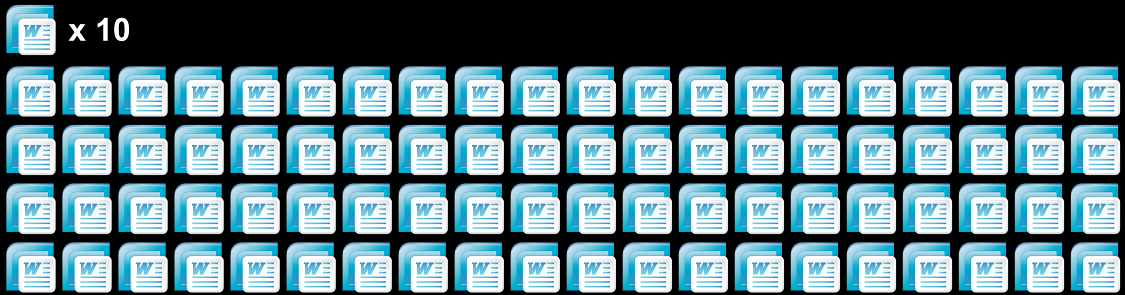
- 2 reviews: 3 deliverables x ~200 releases in 2010
- Minimum of 4 Word documents per release

x 10

realex
The real time payment exchange

Why did I develop Agnitio?

- 2 reviews: 3 deliverables x ~200 releases in 2010
- Minimum of 4 Word documents per release



The real time payment exchange

Why did I develop Agnitio?

- Demonstration: security code review reports

Why did I develop Agnitio?

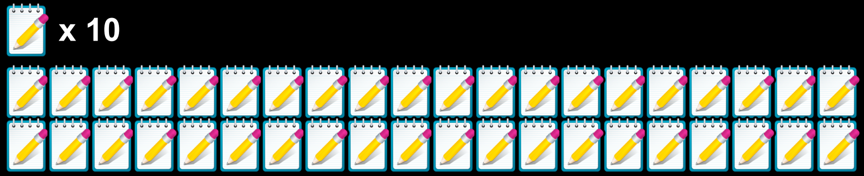
- 2 reviews: 3 deliverables x ~200 releases in 2010
- Note pad file per release with notes, LOC etc

📝 x 10

realex
The real time payment exchange

Why did I develop Agnitio?

- 2 reviews: 3 deliverables x ~200 releases in 2010
- Note pad file per release with notes, LOC etc

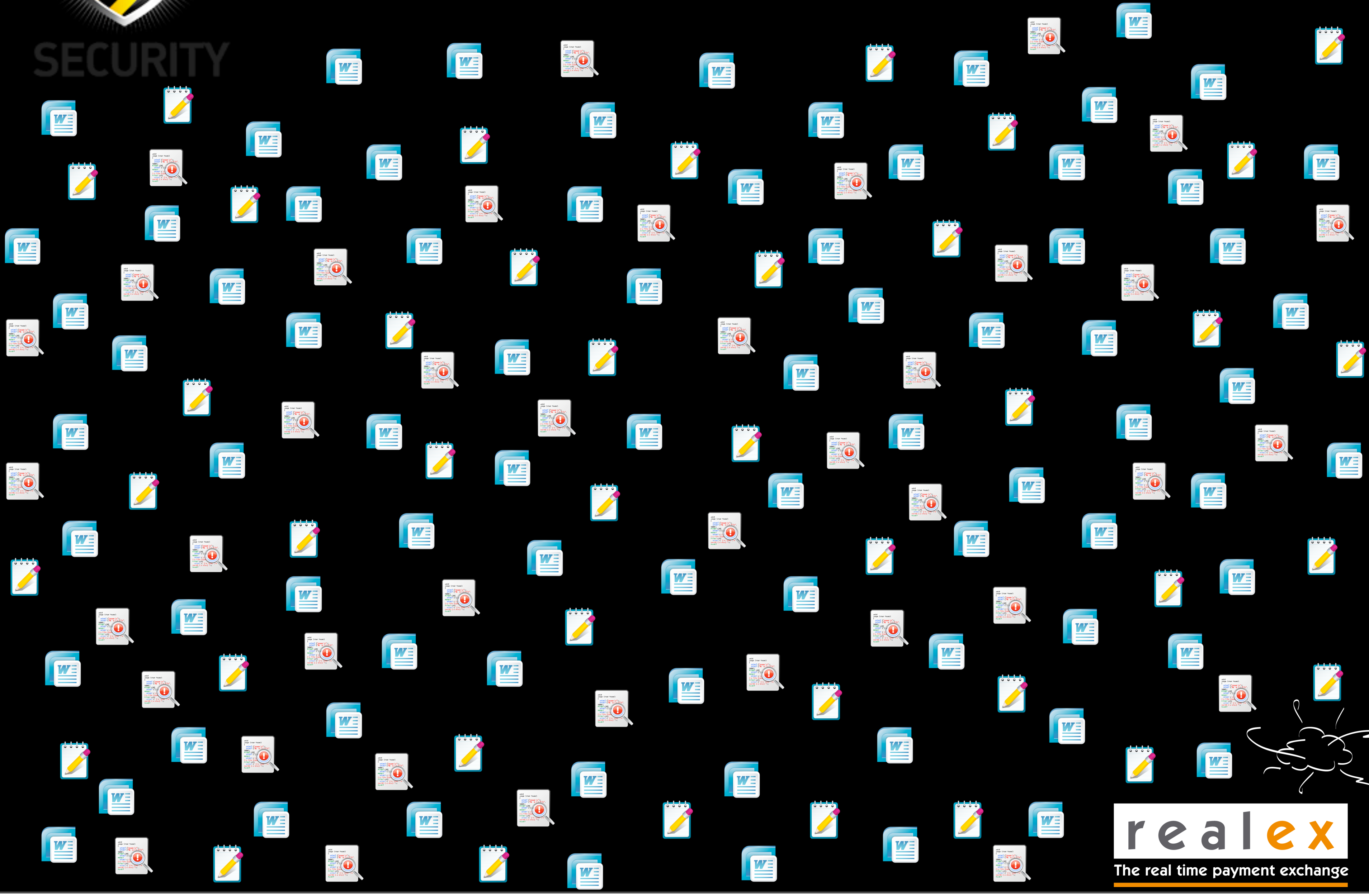


Why did I develop Agnitio?

- Demonstration: application security metrics

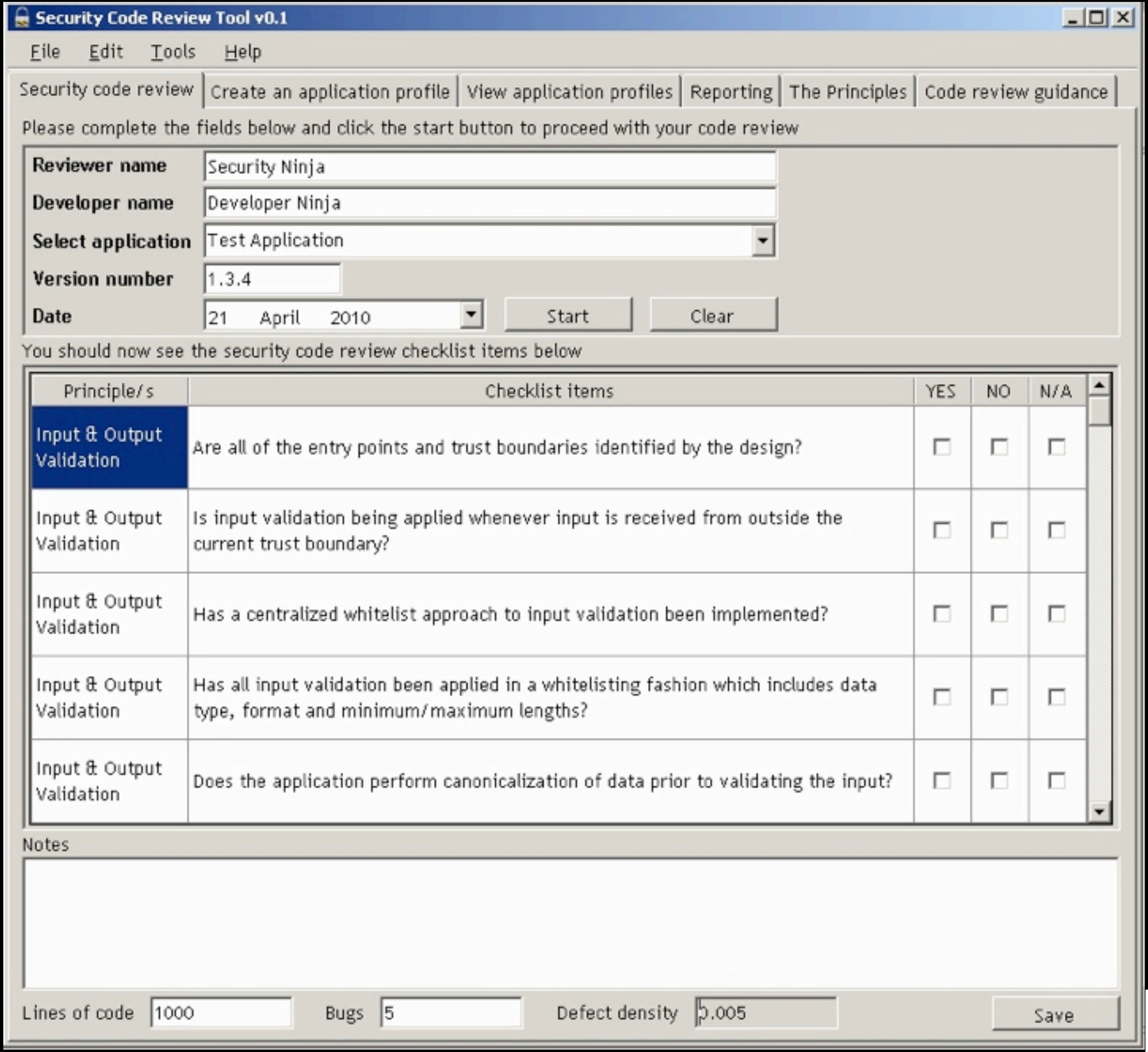
SECURITY

Why did I develop Agnitio?



realex
The real time payment exchange

Why did I develop Agnitio?



Security Code Review Tool v0.1
File Edit Tools Help
Security code review | Create an application profile | View application profiles | Reporting | The Principles | Code review guidance
Start
Clear
Save

Please complete the fields below and click the start button to proceed with your code review

Reviewer name	Security Ninja
Developer name	Developer Ninja
Select application	Test Application
Version number	1.3.4
Date	21 April 2010

You should now see the security code review checklist items below

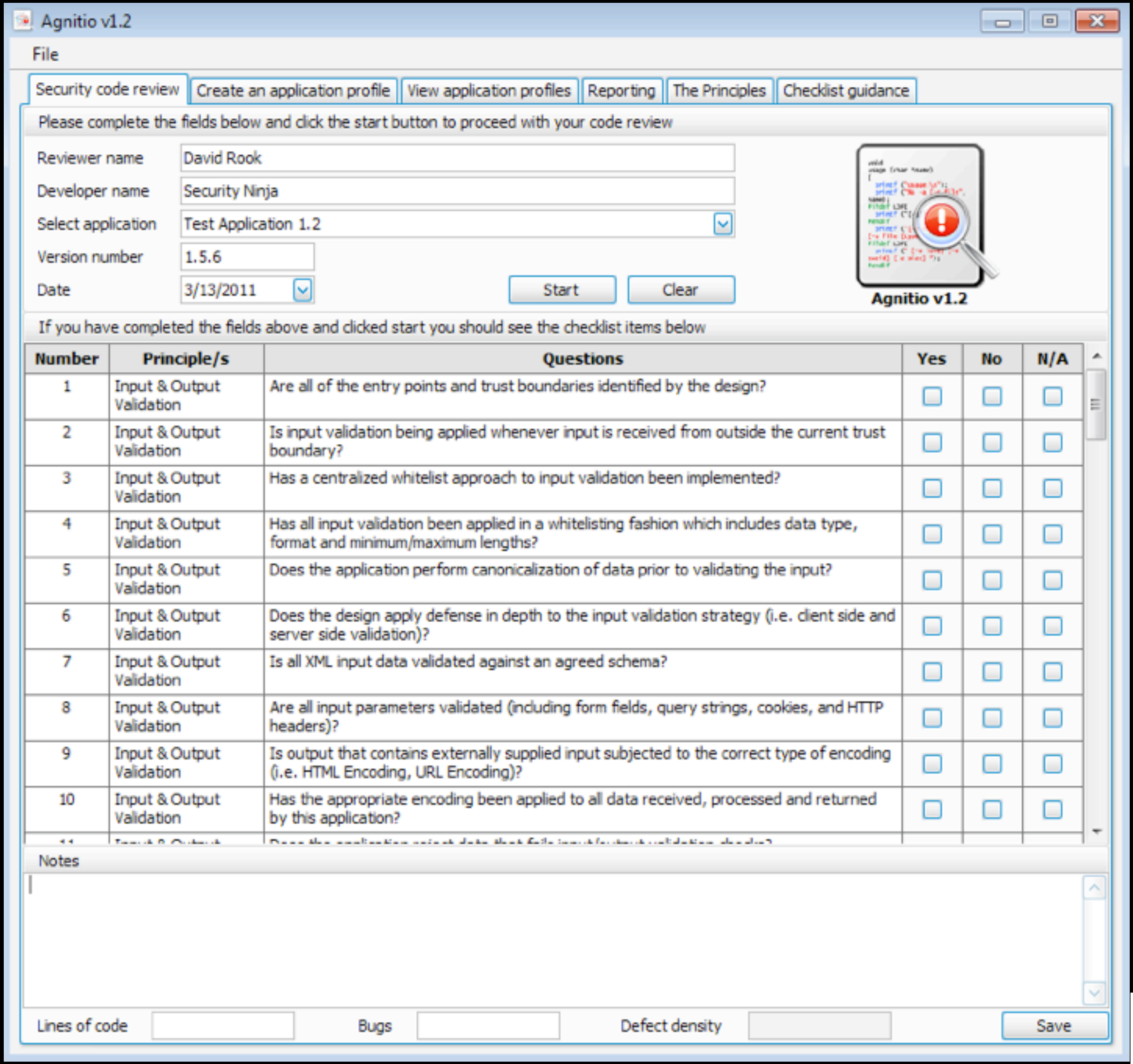
Principle/s	Checklist items	YES	NO	N/A
Input & Output Validation	Are all of the entry points and trust boundaries identified by the design?	☐	☐	☐
Input & Output Validation	Is input validation being applied whenever input is received from outside the current trust boundary?	☐	☐	☐
Input & Output Validation	Has a centralized whitelist approach to input validation been implemented?	☐	☐	☐
Input & Output Validation	Has all input validation been applied in a whitelisting fashion which includes data type, format and minimum/maximum lengths?	☐	☐	☐
Input & Output Validation	Does the application perform canonicalization of data prior to validating the input?	☐	☐	☐

Notes

Lines of code	1000	Bugs	5	Defect density	0.005

realex
The real time payment exchange

Why did I develop Agnitio?



Please complete the fields below and click the start button to proceed with your code review

Reviewer name	David Rook
Developer name	Security Ninja
Select application	Test Application 1.2
Version number	1.5.6
Date	3/13/2011

If you have completed the fields above and clicked start you should see the checklist items below

Number	Principle/s	Questions	Yes	No	N/A
1	Input & Output Validation	Are all of the entry points and trust boundaries identified by the design?	☐	☐	☐
2	Input & Output Validation	Is input validation being applied whenever input is received from outside the current trust boundary?	☐	☐	☐
3	Input & Output Validation	Has a centralized whitelist approach to input validation been implemented?	☐	☐	☐
4	Input & Output Validation	Has all input validation been applied in a whitelisting fashion which includes data type, format and minimum/maximum lengths?	☐	☐	☐
5	Input & Output Validation	Does the application perform canonicalization of data prior to validating the input?	☐	☐	☐
6	Input & Output Validation	Does the design apply defense in depth to the input validation strategy (i.e. client side and server side validation)?	☐	☐	☐
7	Input & Output Validation	Is all XML input data validated against an agreed schema?	☐	☐	☐
8	Input & Output Validation	Are all input parameters validated (including form fields, query strings, cookies, and HTTP headers)?	☐	☐	☐
9	Input & Output Validation	Is output that contains externally supplied input subjected to the correct type of encoding (i.e. HTML Encoding, URL Encoding)?	☐	☐	☐
10	Input & Output Validation	Has the appropriate encoding been applied to all data received, processed and returned by this application?	☐	☐	☐
11	Input & Output	Does the application reject data that fails input/output validation checks?			

Notes

Lines of code | Bugs | Defect density

realex
The real time payment exchange

Agnitio v2.0

- Agnitio deux sera bientôt disponible en Français!
- Automated code analysis module linked to checklist
- Data editor for developer and checklist guidance text
- Checklist and guidance in multiple languages
- Plus lots of user suggested changes!

realex
The real time payment exchange

Agnitio v2.0

- Agnitio v2.0 demonstration

My “shoot for the moon” vision for Agnitio

“we pretty much need a Burp Pro equivalent for Static Analysis – awesome, powerful in the right hands, and completely affordable!”

http://www.securityninja.co.uk/application-security/can-you-implement-static-analysis-without-breaking-the-bank/comment-page-1#comment-9777

realex
The real time payment exchange

Using the principles and Agnitio

- How you can apply the principles approach
  - Download principles documentation from Security Ninja
  - Focus secure development training on code not exploits
  - Use your language/s in all code examples and checklist items
  - Use Agnitio to conduct principles based security code reviews
  - Tie all security findings back to specific principles

realex
The real time payment exchange

www.securityninja.co.uk

http://sourceforge.net/projects/agnitiotool/

@securityninja

/realexninja

/securityninja

/realexninja

QUESTIONS?

www.securityninja.co.uk

http://sourceforge.net/projects/agnitiotool/

@securityninja

/realexninja

/securityninja

/realexninja