**Owasp**

**Pom.xml**

<!-- https://mvnrepository.com/artifact/org.owasp.antisamy/antisamy -->

<dependency>

<groupId>org.owasp.antisamy</groupId>

<artifactId>antisamy</artifactId>

<version>1.5.8</version>

</dependency>

**CharacterEncodingFilter.java**

**@Component**

**@Order(1)**

import java.io.IOException;

import java.util.Enumeration;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

import java.util.Set;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.owasp.validator.html.AntiSamy;

import org.owasp.validator.html.CleanResults;

import org.owasp.validator.html.Policy;

public class CharacterEncodingFilter implements Filter {

@Override

public void destroy() {

System.out.println("destroy calisti ");

}

@SuppressWarnings("unchecked")

@Override

public void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain)

throws IOException, ServletException {

HttpServletRequest request = (HttpServletRequest) req;

HttpServletResponse response = (HttpServletResponse) resp;

request.setCharacterEncoding("UTF8");

response.setCharacterEncoding("UTF8");

chain.doFilter(request, response);

request.setCharacterEncoding("UTF8");

response.setCharacterEncoding("UTF8");

// xss control

try {

Policy policy = Policy.getInstance(CharacterEncodingFilter.class.getResourceAsStream("/antisamy-slashdot-1.4.4.xml"));

AntiSamy sanitizer = new AntiSamy(policy);

Map m = req.getParameterMap();

Set s = m.entrySet();

Iterator it = s.iterator();

while(it.hasNext()){

Map.Entry<String,String[]> entry = (Map.Entry<String,String[]>)it.next();

String key = entry.getKey();

String[] value = entry.getValue();

if(value.length>0){

for (int i = 0; i < value.length; i++) {

System.out.println("val : " + value[i].toString());

CleanResults scanned = sanitizer.scan(value[i].toString());

int errors = scanned.getNumberOfErrors(); // Kural ihlali sayısı

List<String> errorMsg = scanned.getErrorMessages(); // İhlal nedenleri

String sanitized = scanned.getCleanHTML(); // Temizlenmiş çıktı

System.out.println("Temiz çıktı: "+sanitized);

System.out.println("İhlal sayısı: "+errors);

System.out.println("İhlal nedenleri: "+errorMsg);

String ipAddress = request.getHeader("X-FORWARDED-FOR");

if (ipAddress == null) {

ipAddress = request.getRemoteAddr();

System.out.println("ip Address : " + ipAddress);

}

System.out.println("All Header : " + getRequestHeadersInMap(request));

}

}

}

} catch (Exception e) {

System.err.println("req hatası : " + e);

}

}

private Map<String, String> getRequestHeadersInMap(HttpServletRequest request) {

Map<String, String> result = new HashMap<String, String>();

Enumeration headerNames = request.getHeaderNames();

while (headerNames.hasMoreElements()) {

String key = (String) headerNames.nextElement();

String value = request.getHeader(key);

result.put(key, value);

}

return result;

}

@Override

public void init(FilterConfig arg0) throws ServletException {

// TODO Auto-generated method stub

}

}

**antisamy-slashdot-1.4.4.xml**

<?xml version="1.0" encoding="ISO-8859-1"?>

<!--

W3C rules retrieved from:

http://www.w3.org/TR/html401/struct/global.html

-->

<!--

Slashdot allowed tags taken from "Reply" page:

<b> <i> <p> <br> <a> <ol> <ul> <li> <dl> <dt> <dd> <em> <strong> <tt> <blockquote> <div> <ecode> <quote>

-->

<anti-samy-rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="antisamy.xsd">

<directives>

<directive name="omitXmlDeclaration" value="true"/>

<directive name="omitDoctypeDeclaration" value="true"/>

<directive name="maxInputSize" value="5000"/>

<directive name="useXHTML" value="true"/>

<directive name="formatOutput" value="true"/>

<directive name="embedStyleSheets" value="false"/>

</directives>

<common-regexps>

<!--

From W3C:

This attribute assigns a class name or set of class names to an

element. Any number of elements may be assigned the same class

name or names. Multiple class names must be separated by white

space characters.

-->

<regexp name="htmlTitle" value="[\p{L}\p{N}\s\-\_',:\[\]!\./\\\(\)&amp;]\*"/> <!-- force non-empty with a '+' at the end instead of '\*' -->

<regexp name="onsiteURL" value="([\p{L}\p{N}\\/\.\?=\#&amp;;\-\_~]+|\#(\w)+)"/>

<regexp name="offsiteURL" value="(\s)\*((ht|f)tp(s?)://|mailto:)[\p{L}\p{N}]+[~\p{L}\p{N}\p{Zs}\-\_\.@\#\$%&amp;;:,\?=/\+!\(\)]\*(\s)\*"/>

</common-regexps>

<!--

Tag.name = a, b, div, body, etc.

Tag.action = filter: remove tags, but keep content, validate: keep content as long as it passes rules, remove: remove tag and contents

Attribute.name = id, class, href, align, width, etc.

Attribute.onInvalid = what to do when the attribute is invalid, e.g., remove the tag (removeTag), remove the attribute (removeAttribute), filter the tag (filterTag)

Attribute.description = What rules in English you want to tell the users they can have for this attribute. Include helpful things so they'll be able to tune their HTML

-->

<!--

Some attributes are common to all (or most) HTML tags. There aren't many that qualify for this. You have to make sure there's no

collisions between any of these attribute names with attribute names of other tags that are for different purposes.

-->

<common-attributes>

<attribute name="lang" description="The 'lang' attribute tells the browser what language the element's attribute values and content are written in">

<regexp-list>

<regexp value="[a-zA-Z]{2,20}"/>

</regexp-list>

</attribute>

<attribute name="title" description="The 'title' attribute provides text that shows up in a 'tooltip' when a user hovers their mouse over the element">

<regexp-list>

<regexp name="htmlTitle"/>

</regexp-list>

</attribute>

<attribute name="href" onInvalid="filterTag">

<regexp-list>

<regexp name="onsiteURL"/>

<regexp name="offsiteURL"/>

</regexp-list>

</attribute>

<attribute name="align" description="The 'align' attribute of an HTML element is a direction word, like 'left', 'right' or 'center'">

<literal-list>

<literal value="center"/>

<literal value="left"/>

<literal value="right"/>

<literal value="justify"/>

<literal value="char"/>

</literal-list>

</attribute>

</common-attributes>

<!--

This requires normal updates as browsers continue to diverge from the W3C and each other. As long as the browser wars continue

this is going to continue. I'm not sure war is the right word for what's going on. Doesn't somebody have to win a war after

a while?

-->

<global-tag-attributes>

<attribute name="title"/>

<attribute name="lang"/>

</global-tag-attributes>

<tags-to-encode>

<tag>g</tag>

<tag>grin</tag>

</tags-to-encode>

<tag-rules>

<!-- Tags related to JavaScript -->

<tag name="script" action="remove"/>

<tag name="noscript" action="remove"/>

<!-- Frame & related tags -->

<tag name="iframe" action="remove"/>

<tag name="frameset" action="remove"/>

<tag name="frame" action="remove"/>

<tag name="noframes" action="remove"/>

<!-- CSS related tags -->

<tag name="style" action="remove"/>

<!-- All reasonable formatting tags -->

<tag name="p" action="validate">

<attribute name="align"/>

</tag>

<tag name="div" action="validate"/>

<tag name="i" action="validate"/>

<tag name="b" action="validate"/>

<tag name="em" action="validate"/>

<tag name="blockquote" action="validate"/>

<tag name="tt" action="validate"/>

<tag name="strong" action="validate"/>

<tag name="br" action="truncate"/>

<!-- Custom Slashdot tags, though we're trimming the idea of having a possible mismatching end tag with the endtag="" attribute -->

<tag name="quote" action="validate"/>

<tag name="ecode" action="validate"/>

<!-- Anchor and anchor related tags -->

<tag name="a" action="validate">

<attribute name="href" onInvalid="filterTag"/>

<attribute name="nohref">

<literal-list>

<literal value="nohref"/>

<literal value=""/>

</literal-list>

</attribute>

<attribute name="rel">

<literal-list>

<literal value="nofollow"/>

</literal-list>

</attribute>

</tag>

<!-- List tags -->

<tag name="ul" action="validate"/>

<tag name="ol" action="validate"/>

<tag name="li" action="validate"/>

</tag-rules>

<!-- No CSS on Slashdot posts -->

<css-rules>

</css-rules>

</anti-samy-rules>