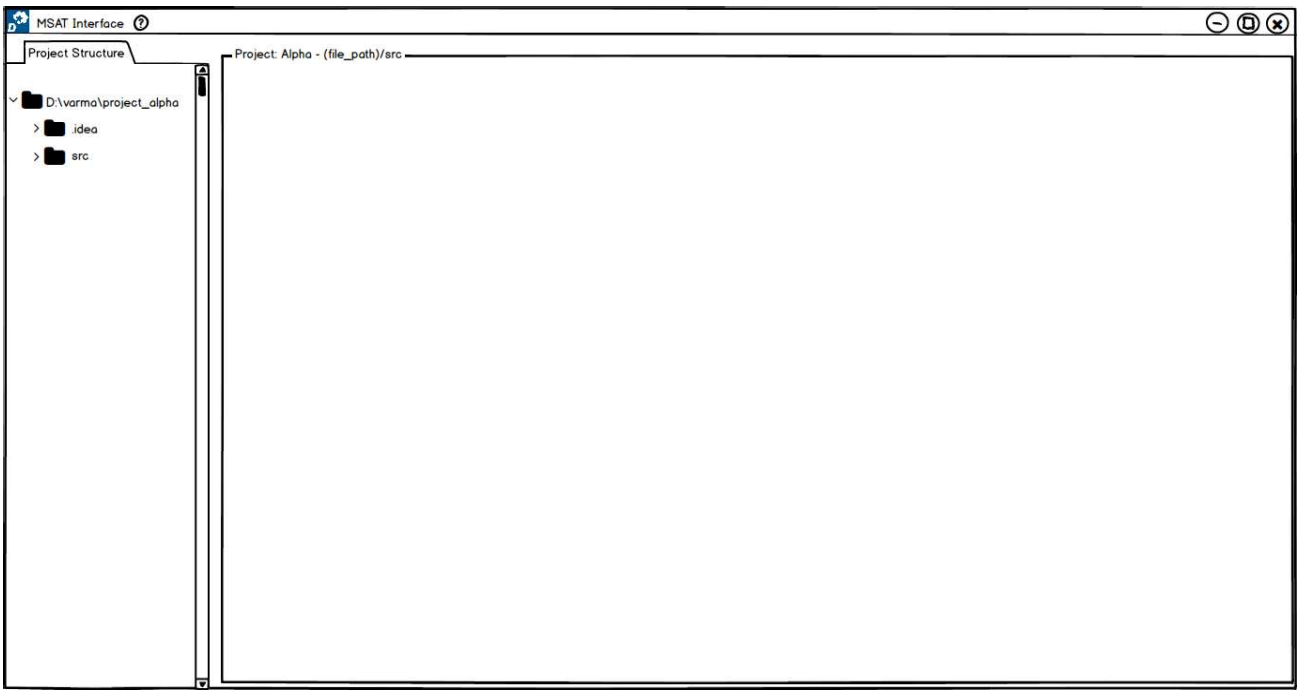
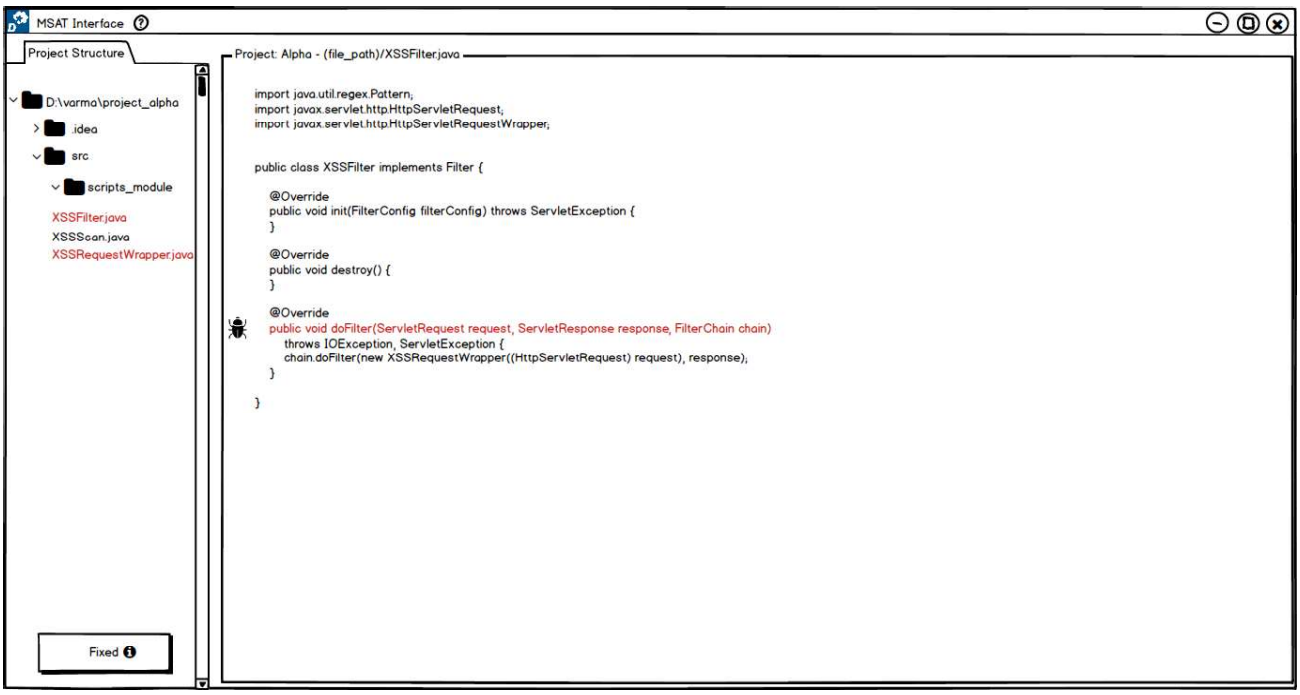
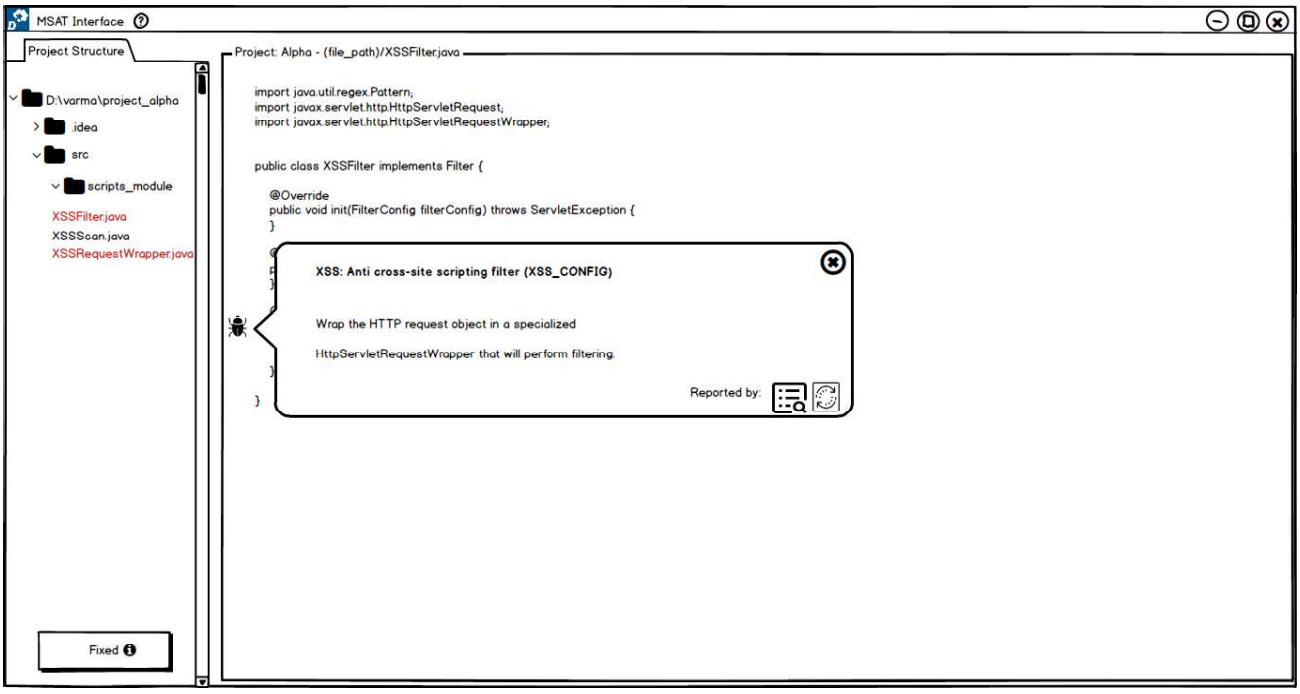

MSAT Interface
Project Structure
Project: Alpha - (file_path)/src
D:\varma\project_alpha
.idea
src


MSAT Interface
Project Structure
Project: Alpha - (file_path)/src
D:\varma\project_alpha
.idea
src
scripts_module


MSAT Interface
Project Structure
Project: Alpha - (file_path)/src
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java

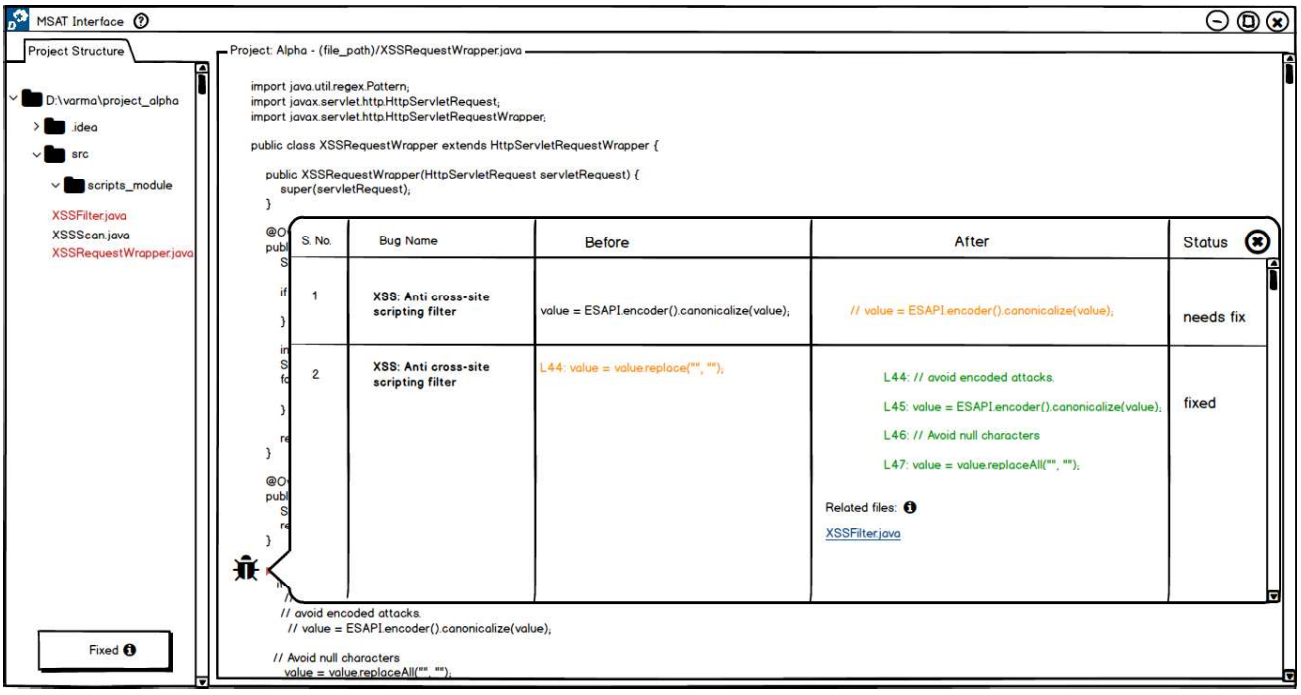

MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSFilter.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;
public class XSSFilter implements Filter {
@Override
public void init(FilterConfig filterConfig) throws ServletException {
}
@Override
public void destroy() {
}
@Override
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
throws IOException, ServletException {
chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
}
}

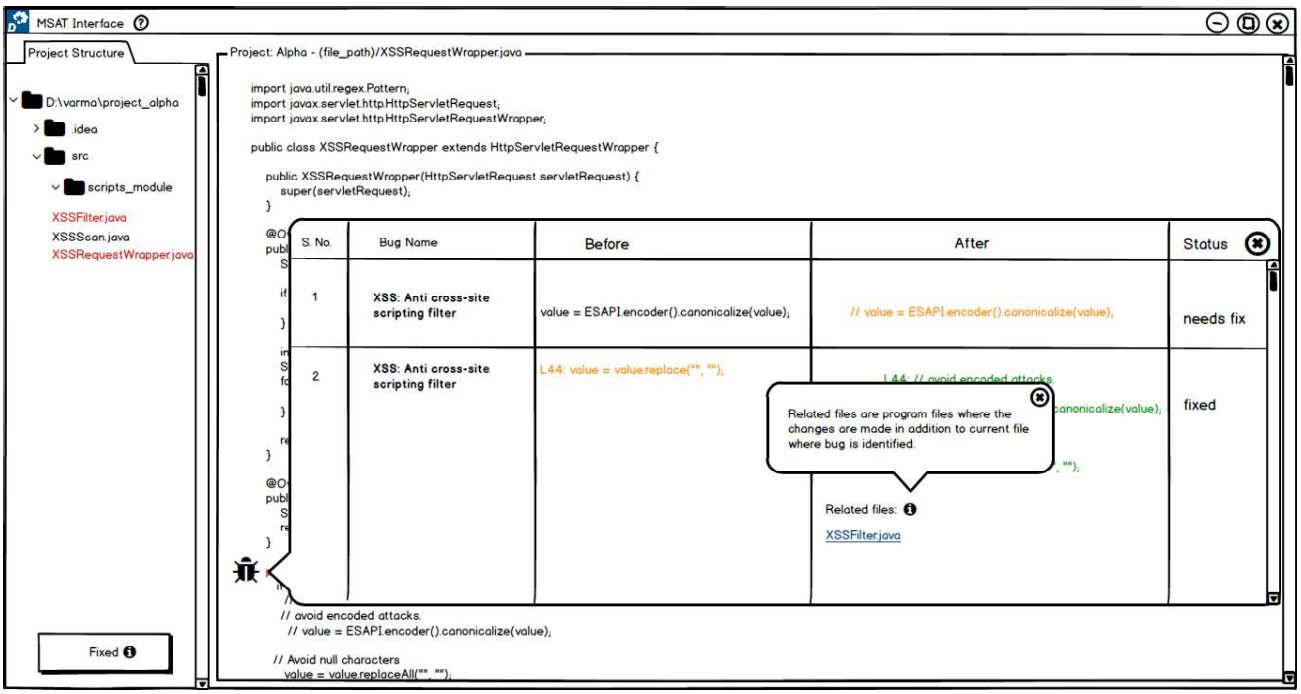

MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSFilter.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;
public class XSSFilter implements Filter {
@Override
public void init(FilterConfig filterConfig) throws ServletException {
}
XSS: Anti cross-site scripting filter (XSS_CONFIG)
Wrap the HTTP request object in a specialized
HttpServletRequestWrapper that will perform filtering.
Reported by:
}

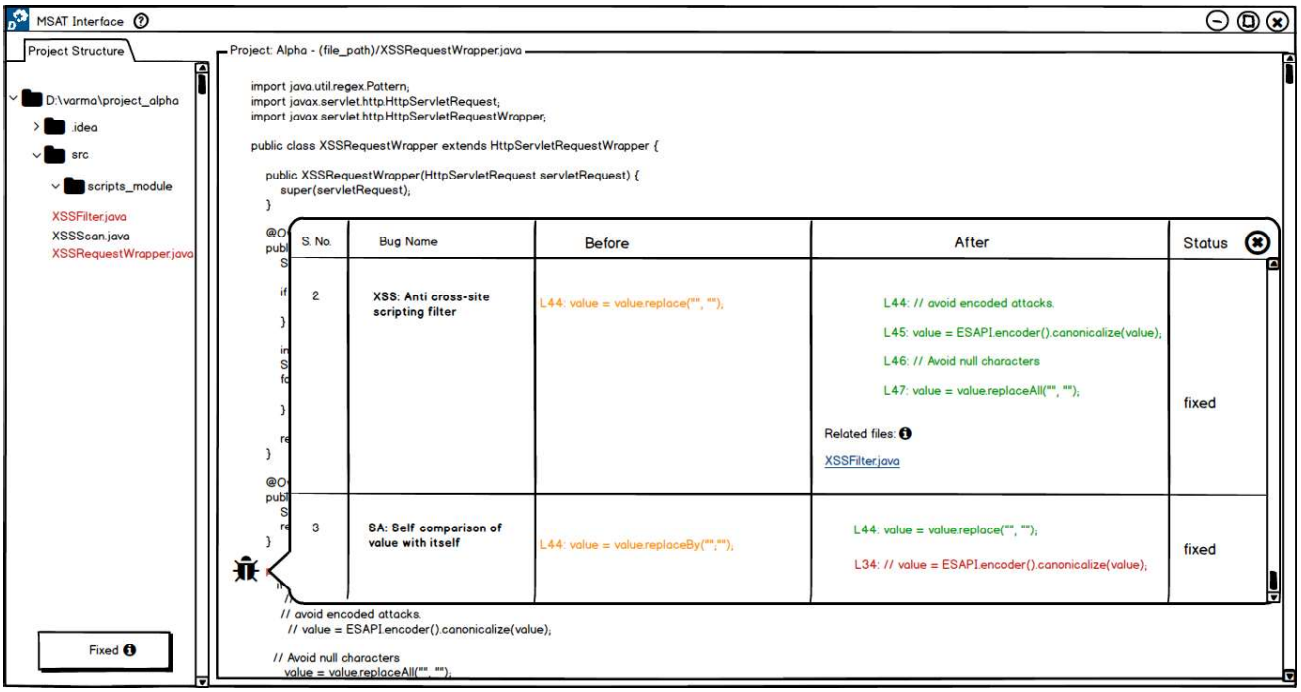

MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSRequestWrapper.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;
public class XSSRequestWrapper extends HttpServletRequestWrapper {
public XSSRequestWrapper(HttpServletRequest servletRequest) {
super(servletRequest);
}
@Override
public String[] getParameterValues(String parameter) {
String[] values = super.getParameterValues(parameter);
if (values == null) {
return null;
}
int count = values.length;
String[] encodedValues = new String[count];
for (int i = 0; i < count; i++) {
encodedValues[i] = stripXSS(values[i]);
}
return encodedValues;
}
@Override
public String getHeader(String name) {
String value = super.getHeader(name);
return stripXSS(value);
}
private String stripXSS(String value) {
if (value != null) {
// NOTE: It's highly recommended to use the ESAPI library and uncomment the following line to
// avoid encoded attacks.
// value = ESAPI.encoder().canonicalize(value);
// Avoid null characters
value = value.replaceAll("" "");

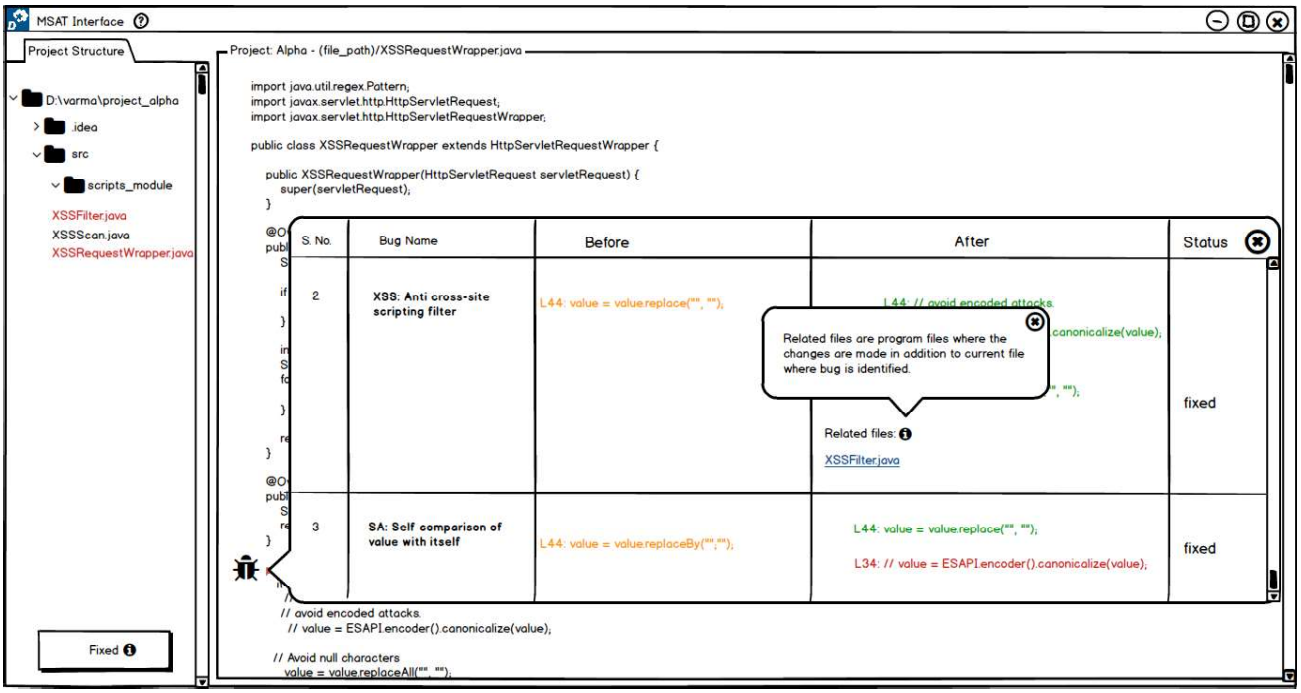

MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSRequestWrapper.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
return encodedValues;
}
@Override
public String getParameter(String parameter) {
String value = super.getParameter(parameter);
return stripXSS(value);
}
@Override
public String getHeader(String name) {
String value = super.getHeader(name);
return stripXSS(value);
}
private String stripXSS(String value) {
if (value != null) {
// NOTE: It's highly recommended to use the ESAPI library and uncomment the following line to
// avoid encoded attacks.
// value = ESAPI.encoder().canonicalize(value);
// Avoid null characters
value = value.replaceAll("", "");
// Avoid anything between script tags
Pattern scriptPattern = Pattern.compile("<script>(.*?)</script>", Pattern.CASE_INSENSITIVE);
value = scriptPattern.matcher(value).replaceAll("");
// Avoid anything in a src='...' type of expression
scriptPattern = Pattern.compile("src[r]=[]\\'(.*?)\\'", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
value = scriptPattern.matcher(value).replaceAll("");
scriptPattern = Pattern.compile("src[]*=[\]*\\"(.*?)\\"", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
value = scriptPattern.matcher(value).replaceAll("");
// Remove any lonesome </script> tag
scriptPattern = Pattern.compile("</script>", Pattern.CASE_INSENSITIVE);
value = scriptPattern.matcher(value).replaceAll("");


MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSRequestWrapper.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
// Avoid expression(...) expressions
scriptPattern = Pattern.compile("expression\((.*?)\)", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
value = scriptPattern.matcher(value).replaceAll("");
// Avoid javascript:... expressions
scriptPattern = Pattern.compile("javascript:", Pattern.CASE_INSENSITIVE);
value = scriptPattern.matcher(value).replaceAll("");
// Avoid vbscript:... expressions
scriptPattern = Pattern.compile("vbscript:", Pattern.CASE_INSENSITIVE);
value = scriptPattern.matcher(value).replaceAll("");
// Avoid onload= expressions
scriptPattern = Pattern.compile("onload(.*?)=", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
value = scriptPattern.matcher(value).replaceAll("");
}
return value;
}
}


MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSRequestWrapper.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;
public class XSSRequestWrapper extends HttpServletRequestWrapper {
public XSSRequestWrapper(HttpServletRequest servletRequest) {
super(servletRequest);
}
S. No.
Bug Name
Before
After
Status
1
XSS: Anti cross-site scripting filter
value = ESAPI.encoder().canonicalize(value);
// value = ESAPI.encoder().canonicalize(value);
needs fix
2
XSS: Anti cross-site scripting filter
L44: value = value.replace("", "");
L44: // avoid encoded attacks.
L45: value = ESAPI.encoder().canonicalize(value);
L46: // Avoid null characters
L47: value = value.replaceAll("", "");
Related files:
XSSFilter.java
fixed
// avoid encoded attacks.
// value = ESAPI.encoder().canonicalize(value);
// Avoid null characters
value = value.replaceAll("", "");


MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSRequestWrapper.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
S. No.
Bug Name
Before
After
Status
1
XSS: Anti cross-site scripting filter
value = ESAPI.encoder().canonicalize(value);
// value = ESAPI.encoder().canonicalize(value);
needs fix
2
XSS: Anti cross-site scripting filter
L44: value = value.replace("", "");
fixed
Related files are program files where the changes are made in addition to current file where bug is identified.
Related files:
XSSFilter.java


MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSRequestWrapper.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
S. No.
Bug Name
Before
After
Status
2
XSS: Anti cross-site scripting filter
L44: value = value.replace("", "");
L44: // avoid encoded attacks.
L45: value = ESAPI.encoder().canonicalize(value);
L46: // Avoid null characters
L47: value = value.replaceAll("", "");
fixed
Related files:
XSSFilter.java
3
SA: Self comparison of value with itself
L44: value = value.replaceBy("","");
L44: value = value.replace("", "");
L34: // value = ESAPI.encoder().canonicalize(value);
fixed


MSAT Interface
Project Structure
Project: Alpha - (file_path)/XSSRequestWrapper.java
D:\varma\project_alpha
.idea
src
scripts_module
XSSFilter.java
XSSScan.java
XSSRequestWrapper.java
Fixed
S. No.
Bug Name
Before
After
Status
2
XSS: Anti cross-site scripting filter
L44: value = value.replace("", "");
fixed
Related files are program files where the changes are made in addition to current file where bug is identified.
Related files:
XSSFilter.java
3
SA: Self comparison of value with itself
L44: value = value.replaceBy("","");
L44: value = value.replace("", "");
L34: // value = ESAPI.encoder().canonicalize(value);
fixed