MSAT Interface ⑦

🔍 search

Filters

☑ Select All ☐ Deselect All

+24 ⬆ XSS_CONFIG [click here to select bug](#)
⬇ Wrap the HTTP request object in a specialized HttpServletRequestWrapper that will perform filtering.

[tool1](#)   [tool2](#)   [XSS](#)

+18 ⬆ FI_EMPTY
⬇ FI: Empty finalizer should be deleted

[tool3](#)   [FI](#)

+36 ⬆ EQ_CHECK
⬇ Equal checks for incompatible operand

[tool2](#)   [EQ](#)

+44 ⬆ CO_SELF
This class defines a covariant version of compareTo().
⬇ To correctly override the compareTo() method in the Comparable interface, the parameter of compareTo() must have type java.lang.Object.

[tool5](#)   [CO](#)

+8 ⬆ XSS_REQUEST
⬇ This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability.

[tool5](#)   [XSS](#)

+20 ⬆ BC_EQUALS
The equals(Object o) method shouldn't make any assumptions about the type of o.
⬇ It should simply return false if o is not the same type as this.

[tool6](#)   [BC](#)

+12 ⬆ BIT_CHECK
⬇ Check for sign of bitwise operation

☑ tool1
☑ tool2
☑ tool3
☑ tool4
☑ tool5
☑ tool6
☑ tool7
☑ tool8
☑ tool9
☑ tool10

Trending

Priority

MSAT Interface ⑦

Project: Alpha

🔍 search

Filters

☑ Select All ☐ Deselect All

☑ tool1

☑ tool2

☑ tool3

☑ tool4

☑ tool5

☑ tool6

☑ tool7

☑ tool8

☑ tool9

☑ tool10

+33 ⬆ BIT_CHECK
⬇ Check for sign of bitwise operation
              tool8   BIT

+17 ⬆ CN_CLONE
⬇ clone method does not call super.clone()
              tool8   CN

+34 ⬆ DE_EXCEPTION
⬇ Method might drop exception
              tool5   DE

+40 ⬆ DMI_RANDOM
⬇ Random object created and used only once
              tool7   DMI

-2 ⬆ EQ_EQUALS
⬇ Equals checks for incompatible operand
              tool2   EQ

+22 ⬆ IJU_TEST
⬇ Class is a JUnit TestCase but has not implemented any test methods
              tool3   IJU

+10 ⬆ IL_LOOP
⬇ This loop doesn't seem to have a way to terminate (other than by perhaps throwing an exception).

Trending

Priority

Project: Alpha

Filters

Q search

☑ Select All ☐ Deselect All

☑ tool1

+24 ⬆️⬇️ EQ_EQUALS
Equals checks for incompatible operand

tool2 EQ

☑ tool2

+18 ⬆️⬇️ IJU_TEST
Class is a JUnit TestCase but has not implemented any test methods

tool3 IJU

☑ tool3

+36 ⬆️⬇️ IL_LOOP
This loop doesn't seem to have a way to terminate (other than by perhaps throwing an exception).

tool7 IL

☑ tool4

☑ tool5

+44 ⬆️⬇️ CI_FINAL
Class is final but declares protected field

tool2 CI

☑ tool6

+8 ⬆️⬇️ SQL_CONSTANT
Method attempts to access a result set field with index 0

tool4 SQL

☑ tool7

☑ tool8

+20 ⬆️⬇️ XSS_SERVLET
Servlet reflected cross site scripting vulnerability in error page

tool6 XSS

☑ tool9

☑ tool10

Trending

+12 ⬆️⬇️ DM_EXIT
Method invokes System.exit(...)

tool5 DM

Priority

Project: Alpha

Filters

🔍 search

☑ Select All ☐ Deselect All

☑ tool1

NEW ⬆ XSS_CONFIG click here to select bug
⬇ Wrap the HTTP request object in a specialized HttpServletRequestWrapper that will perform filtering.

tool1 tool2 XSS

☑ tool2

NEW ⬆ FI_EMPTY
⬇ FI: Empty finalizer should be deleted

tool3 FI

☑ tool3

NEW ⬆ EQ_CHECK
⬇ Equal checks for incompatible operand

tool2 EQ

☑ tool4

NEW ⬆ CO_SELF
⬇ This class defines a covariant version of compareTo().
To correctly override the compareTo() method in the Comparable interface, the parameter of compareTo() must have type java.lang.Object.

tool5 CO

☑ tool5

☑ tool6

NEW ⬆ XSS_REQUEST
⬇ This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability.

tool5 XSS

☑ tool7

+2 ⬆ BC_EQUALS
⬇ The equals(Object o) method shouldn't make any assumptions about the type of o.
It should simply return false if o is not the same type as this.

tool6 BC

☑ tool8

☑ tool9

☑ tool10

Trending

+7 ⬆ BIT_CHECK
⬇ Check for sign of bitwise operation

Priority

Project: Alpha

Filters

🔍 search

☑ Select All ☐ Deselect All

☑ tool1

☑ tool2

☑ tool3

☑ tool4

☑ tool5

☑ tool6

☑ tool7

☑ tool8

☑ tool9

☑ tool10

+46
⬆
⬇
                    tool8   BIT

+40
⬆
⬇

BIT_CHECK
Check for sign of bitwise operation
                    tool8   CN

+39
⬆
⬇

CN_CLONE
clone method does not call super.clone()
                    tool5   DE

+35
⬆
⬇

DE_EXCEPTION
Method might drop exception

+28
⬆
⬇

DMI_RANDOM
Random object created and used only once
                    tool2   EQ

+24
⬆
⬇

EQ_EQUALS
Equals checks for incompatible operand
                    tool3   IJU

+20
⬆
⬇

IJU_TEST
Class is a JUnit TestCase but has not implemented any test methods.

Trending

Priority

IL_LOOP
This loop doesn't seem to have a way to terminate (other than by perhaps throwing an exception).

MSAT Interface ⊘

🔍 search

Filters

☐ Select All   ☐ Deselect All

☑ tool1

☐ tool2

☐ tool3

☐ tool4

☐ tool5

☐ tool6

☐ tool7

☐ tool8

☐ tool9

☐ tool10

+24 ⬆ BT1
⬇ description line1
       tool1 EQ tool3

+18 ⬆ BT1
⬇ description line1
       tool1 IJU

+36 ⬆ BT1
⬇ description line1
       tool1 IL tool7

+44 ⬆ BT1
⬇ description line1
       tool1 CI

+8 ⬆ BT1
⬇ description line1
       tool1 SQL

+20 ⬆ BT1
⬇ description line1
       tool1 XSS

+12 ⬆ BT1
⬇ description line1
       tool1 DM

Trending

Priority

MSAT Interface ⑦

— Project: Alpha —

Filters

search

☐ Select All   ☐ Deselect All

+2 ⬆ BT1
⬇ description line1

tool1 tool3 IJU

+19 ⬆ BT1
⬇ description line1

tool1 EQ

+26 ⬆ BT1
⬇ description line1

tool1 CI

+14 ⬆ BT1
⬇ description line1

tool1 tool7 IL

+27 ⬆ BT1
⬇ description line1

tool1 DM

+30 ⬆ BT1
⬇ description line1

tool1 SQL

+8 ⬆ BT1
⬇ description line1

tool1 XSS

☑ tool1

☐ tool2

☐ tool3

☐ tool4

☐ tool5

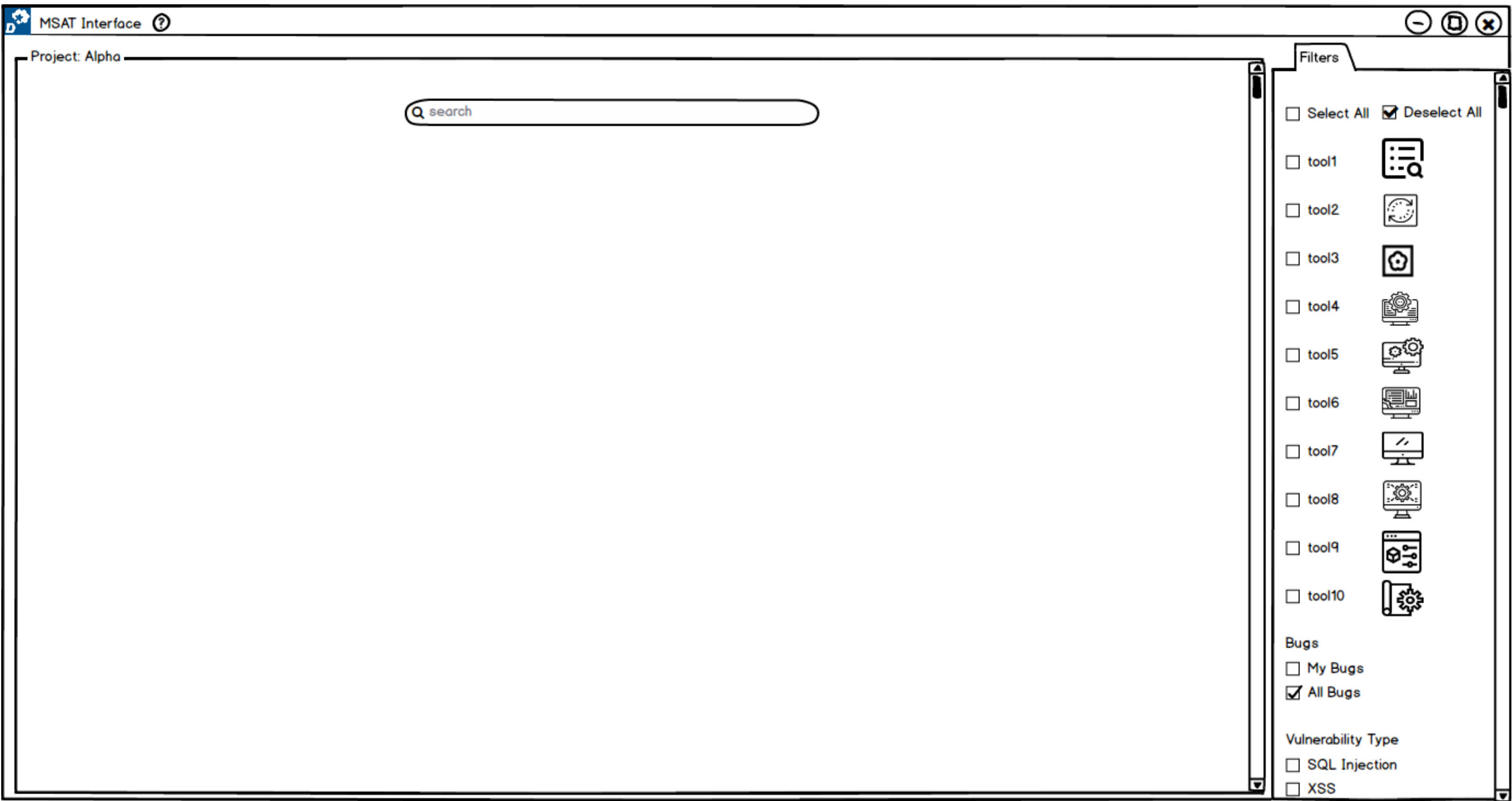☐ tool6

☐ tool7

☐ tool8

☐ tool9

☐ tool10

Trending

Priority

Project: Alpha

search

Filters

☐ Select All ☑ Deselect All

☐ tool1

☐ tool2

☐ tool3

☐ tool4

☐ tool5

☐ tool6

☐ tool7

☐ tool8

☐ tool9

☐ tool10

Bugs

☐ My Bugs

☑ All Bugs

Vulnerability Type

☐ SQL Injection

☐ XSS

Project: Alpha - Selected Bug Description

XSS: Anti cross-site scripting filter (XSS_CONFIG) ⊗

You should configure it as the first filter in your chain (web.xml) and it's generally a good idea to let it catch every request made to your site.
The actual implementation should consist of two classes, the actual filter is quite simple, it wraps the HTTP request object in a specialized HttpServletRequestWrapper that will perform the filtering.

he wrapper overrides the getParameterValues(), getParameter() and getHeader() methods to execute the filtering before returning the desired field to the caller. The actual XSS checking and striping is performed in the stripXSS() private method.

Suggested Quick Fixes:

Apply

Filters

Tool

☑ tool1

☑ tool2

☑ tool3

☑ tool4

☑ tool5

☑ tool6

☑ tool7

☑ tool8

☑ tool9

☑ tool10

Bugs

☐ My Bugs
☑ All Bugs

Vulnerability Type

☐ SQL Injection

Project Structure

D:\varma\project_alpha
- .idea
- src
  - scripts_module
    - XSSFilter.java
    - XSSScan.java
    - XSSRequestWrapper.java

Fixed ⓘ

```java
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);
    }

}
```