S2-013

Summary

A vulnerability, present in the *includeParams* attribute of the *URL* and *Anchor* Tag, allows remote command execution

Who should read this	All Struts 2 developers
Impact of vulnerability	Remote command execution
Maximum security rating	Critical
Recommendation	Developers should immediately upgrade to at least Struts 2.3.14.2
Affected Software	Struts 2.0.0 - Struts 2.3.14.1
Reporter	The Struts Team
CVE Identifier	CVE-2013-1966

Problem

Both the *s:url* and *s:a* tag provide an *includeParams* attribute.

The main scope of that attribute is to understand whether includes http request parameter or not.

The allowed values of includeParams are:

1. *none* - include no parameters in the URL (default)
2. *get* - include only GET parameters in the URL
3. *all* - include both GET and POST parameters in the URL

A request that included a specially crafted request parameter could be used to inject arbitrary OGNL code into the stack, afterward used as request parameter of an *URL* or *A* tag , which will cause a further evaluation.

The second evaluation happens when the URL/A tag tries to resolve every parameters present in the original request.
This lets malicious users put arbitrary OGNL statements into any request parameter (not necessarily managed by the code) 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. Open HelloWorld.jsp present in the Struts Blank App and add to one of the url/a tag the following parameter:

   ```
   includeParams="all"
   ```

   Such that the line will be something look like this:

   ```
   <s:url id="url" action="HelloWorld" includeParams="all">
   ```

   (it works also with *includeParams="get"*).
2. Run struts2-blank app
3. Open the url: http://localhost:8080/example/HelloWorld.action?fakeParam=%25%7B(%23_memberAccess%5B'allowStaticMethodAccess'%5D%3Dtrue)(%23context%5B'xwork.MethodAccessor.denyMethodExecution'%5D%3Dfalse)(%23writer%3D%40org.apache.struts2.ServletActionContext%40getResponse().getWriter()%2C%23writer.println('hacked')%2C%23writer.close())%7D
   (this is the shortened version http://goo.gl/lhlTl)

As you will notice, in this case, there is no way to escape/sanitize the fakeParam, since it's not an expected parameter.

Solution

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

Backward Compatibility

In case you need to restore the old behavior, you need to define the following constant, inside your struts configuration (**use it at your own risk**).

```
<constant name="struts.ognl.enableOGNLEvalExpression" value="true" />
```

Please, ensure that:

1. there are no *includeParams* with "all" or "get" value
2. every parameter which is declared inside the *u* or *a* tag come from a sanitized input.

It is strongly recommended to upgrade to at least Struts 2.3.14.2, which contains the corrected OGNL and XWork library.